DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CURRICULUM PLAN OPTIMIZATION WITH RULE BASED GENETIC ALGORITHMS

by
Didem ABİDİN

April, 2013
İZMİR

CURRICULUM PLAN OPTIMIZATION WITH RULE BASED GENETIC ALGORITHMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering Program

by

Didem ABİDİN

April, 2013
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"CURRICULUM PLAN OPTIMIZATION WITH RULE BASED GENETIC ALGORITHMS"** completed by **DİDEM ABİDİN** under supervision of **ASSIST. PROF. ŞEN ÇAKIR** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

ASSIST. PROF. ŞEN ÇAKIR

Supervisor

PROF. DR. HALDUN KARACA	ASSOC. PROF. AYBARS UĞUR
Thesis Committee Member	Thesis Committee Member
ASSOC. PROF. DR. AYLIN KANTARCI	PROF. DR. BIRGUL KUTLU BAYRAKTAR
Examining Committee Member	Examining Committee Member

Prof.Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

First and foremost, I thank to my supervisor of this thesis, Mrs. Çakır for her valuable guidance and advices. She inspired me greatly to work in this project. Her willingness to motivate me contributed tremendously to the thesis. I also thank Mr. Uğur for his advices and cheering words. Besides, I would like to thank my directors and co-workers at Ege University Tire Kutsan Vocational School for their patience because they were enormously tolerant while I deal with my thesis despite our busy curricula. Finally, an honorable mention goes to my family and friends for their understandings and supports on me in completing this thesis. Without helps of the particular that mentioned above, I would face many difficulties while completing the thesis.

Didem ABİDİN

CURRICULUM PLAN OPTIMIZATION WITH RULE BASED GENETIC ALGORITHMS

ABSTRACT

In corporations, accurate planning should be applied to manage the in – service training task within an optimum time period and without hindering the working tempo of the employees. For this reason, it is better to consider the curriculum planning task as a timetabling problem. However, when the timetables are prepared manually, it may turn out to be a complicated and time consuming problem. In this study, an effective solution to the curriculum planning problem by using a rule – based genetic algorithm is put forward. The data, which is used by the fitness function of the GA to obtain the results, is the prerequisite rule set of the modules of the training program. The contribution to the literature is handling the structure of its data set successfully, despite tightly related rules among the modules. The modules of a training material were ranked effectively and while performing the ranking process, parameter tuning for GA was done to determine the best parameter combination of GA. The tests were done for two different amounts of modules. The results were then compared with the suggestion of an expert trainer by using Spearman rank correlation test, which is nonparametric, and the best parameter combination of the GA giving the most similar result to that of the expert's was determined. According to the tests, the results gathered were considered to be 98.53 percent reliable for the smaller size of module ranges (chromosomes) and 97.06 percent reliable for the larger size of module ranges when compared with the corresponding suggested module range. Same tests were repeated with a control data set, having the same characteristics with the first one and two different sizes, and the results verified that same parameter combinations give the same successful module ranges in the same reliability percentages.

Keywords: Genetic algorithm, rule base, curriculum plan optimization, Spearman rank correlation.

KURAL TABANLI GENETİK ALGORİTMALAR İLE EĞİTİM PLANI OPTİMİZASYONU

ÖZ

Şirketlerde, şirket içi eğitim sürecinin optimum sürede ve çalışanların iş temposunu etkilemeden gerçekleştirilmesi için kesin ve hassas bir planlama yapılması gerekmektedir. Bu sebeple bir eğitim planı hazırlanması işlemini bir zaman çizelgeleme problemi olarak ele almak uygun olur. Zaman çizelgeleri elle hazırlandığı zaman karmaşık ve çok zaman alan bir probleme dönüşebilmektedir. Bu çalışmada, kural tabanlı genetik algoritma (GA) kullanılarak eğitim planı hazırlama problemine etkin bir çözüm ortaya konmaktadır. GA'nın uygunluk fonksiyonunun çözüm elde etmek için kullandığı veriler, eğitim programındaki bölümlerin birbirlerine gore ön koşul durumlarını içeren bir kurallar kümesinden oluşmaktadır. Çalışmanın literature katkısı birbirine sıkı kurallarla bağlı modülleri olan bir eğitim materyalinin veri kümesini başarılı bir şekilde işleyebilmesidir. Eğitim materyalinin bölümleri olan modüller etkin bir biçimde sıralanabilmekte ve bu işlem esnasında da sıralama işlemi için kullanılacak en uygun parametre kombinasyonunu tespit etmek üzere parametre uyumlaması yapılmaktadır. Testler iki farklı modül sayısı için gerçekleştirilmiştir. Sonuçlar bir uzman önerisi ile parametrik olmayan Spearman sıra korelasyon testi kullanılarak karşılaştırılmış ve uzman önerisine en yakın sonuç tespit edilmiştir. Buna göre, elde edilen sonuçlar uzman önerisi ile karşılaştırıldığında, küçük boyutlu modül dizilimleri için yüzde 98,53, büyük boyutlu modül dizilimleri için ise yüzde 97,06 oranında "güvenilir" bulunmuştur. Aynı özelliklere sahip ve iki farklı büyüklükte bir kontrol veri grubu ile testler tekrarlanmış ve aynı parametre kombinasyonları ile en başarılı dizilim sonuçlarının alınabildiği doğrulanmıştır.

Anahtar sözcükler: Genetik algoritma, kural tabanı, eğitim planı optimizasyonu, Spearman sıra korelasyonu.

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

In the 1950's the idea of "thinking machines" was stated by Alan Turing and it was predicted that these intelligent machines would play a major part in our lives within a century. Machines are not very capable of thinking as humans yet, however, studies about "intelligence" have become very popular over the years. Artificial intelligence (AI), which is a branch of computer science, aims to understand "intelligence" by developing some computer programs which can behave as an intelligent being. A computer is supposed to behave as a human to be accepted as "intelligent". These intelligent programs are used commonly in every area of our daily lives. For this reason, artificial intelligence has some sub branches, in which the researchers apply some search techniques to solve optimization and scheduling problems of today's world.

For decades, the researchers have dealt with studies about understanding the human brain and its behavior. For this purpose, they tried to simulate the behavior of the brain as a chain of actions and reactions of the neurons. These simulations have been used for both scientific modeling purposes in theoretical approaches and solving practical problems. During these studies, many different sub-branches of artificial intelligence emerged. The methodologies invented as the results of these sub-branches have found many different application areas in people's life. These techniques have mostly been used for solving complex problems, the problems that take too much time to solve manually or the ones that the solution is not very obvious at the beginning. By this means, artificial intelligence permits people to construct solution models to the problems and provides automatic design methods.

Practical problems generally appear in real – world problems rather than isolated laboratory environments. For example in business life, many problems, which need the help of some modeling techniques to be solved, emerge. The companies need to be well equipped in terms of solving problems in order to compete with other companies in the same sector. In order to catch up with the technological improvements and higher life standards, companies use timetables or curriculum plans, which help to determine the workflow of the companies and to make use of the personnel and technical hardware effectively. Also they organize some training programs to make the employees be aware of the innovations in the sector. Since awareness means well trained personnel, some training programs must be used and these must be scheduled well. Therefore, the concept of curriculum planning can be mentioned as a rather popular research area. However, preparing an optimum curriculum plan manually might turn out to be a quite complex and time consuming problem. So many people from different departments of the company have to gather together to find the optimum curriculum plan for their trainees and this is a quite challenging task. In a corporation, it is a must to take the constraints of each component (instructors and classrooms for a school, flight traffic for an airport, nurse rostering or operating room timetable for a hospital (Cardoen, Demeulemeester & Belien, 2010), etc.) as in timetabling problems into consideration. For this reason, it is a good application area for the researchers who work on optimization problems of real life.

Researchers studied on different computer programs with different techniques to find solutions to the daily life problems. In some problems every data used in the program are certain and precise, where in some cases there are uncertain things in the definition of the problem. In 1990's a new notion, Soft Computing (SC), is introduced by Zadeh (1994) and it suggests solutions to the cases where uncertainty occurs. SC inspires from the working principles of the human mind because human mind has always a tolerance for the imprecise and uncertain data. However in hard computing, the analysis of the problem and the model for the solution must be stated precisely. The inputs and the outputs of the program should be defined clearly.

SC optimizes the time and the quality of the solution while solving problems which are unsolvable or difficult to solve with traditional methods. SC is made up of different components like Fuzzy Logic (FL), Neurocomputing (NC), Machine Learning (ML), Evolutionary Computation (EC), Particle Swarm Optimization (PSO) and Probabilistic Reasoning (PR). These methods are considered as complementary for each other rather than being alternatives (Selouani, 2011). This means that these techniques perform better in solving a problem when they are used together. There is a wide application area for SC. These are briefly as follows: Biometrics, bioinformatics, biomedical systems, robotics, vulnerability analysis, character recognition, natural language processing (NLP), multi-objective optimizations, wireless networks, financial time series prediction, image processing, toxicology, machine control, software engineering, information management, picture compression, music, noise removal, data mining and social network analysis (Shukla, Tiwari & Kala, 2010).

In optimization problems, when it is needed to find an optimum solution with the minimum cost, the solution can be generated with soft computing techniques. Another advantage of SC methods is that, it is not needed to specify every detail of the solution model of the system from the beginning (Castillo & Melin, 1996) because they are non-linear systems and are able to approximate to the solution easily than linear models (Castillo, Melin, Kacprzyk & Pedrycz, 2008). For example, timetabling and curriculum planning problems are generally difficult to manage manually or with linear programming solutions. For this reason, some evolutionary algorithms and stochastic search techniques are used while dealing with such complex problems.

As one of the soft computing techniques, Genetic algorithms (GA) are said to be the most appropriate search methodology for optimization problems. They were first suggested by Holland (1975) and developed by Goldberg (1989). The GAs are still being improved since Holland and Goldberg within the same principles' framework. The algorithm basically inspires from the natural selection mechanism of nature, in which the best living things survive and the worst ones die. In other words, it is the simulation of the evolution mechanism of nature in computer environment. The transportation mechanism of the genetic material in living organisms in nature is simulated as a population of individuals and the genetic operators of the GA. As the result of the genetic reproduction mechanisms, the genetic diversity of the individuals causes the algorithm to reach many different possible solutions. GA does not find only one solution to a problem; but instead, finds a solution set, in which all solutions in the set are valid. This means that individuals of the final population should be the ones carrying more qualified genetic material to survive for more number of generations.

GA may be applied to many different application areas like job shop scheduling, circuit design, weather forecast, bidding strategies, prediction of a protein structure, automatic programming, modeling natural immunity systems, understanding behavior of insect colonies, evolution and learning, telecommunication and network design.

With genetic algorithms, Expert Systems (ES) can also be used in solving curriculum planning problems. Expert systems are computer programs, designed to solve real world problems instead of a human expert in a certain subject to make decisions and find solutions to a problem by using its own inference mechanisms and human expertise data (Giarratano & Riley, 2004). Expert systems are also called *knowledge based systems (KBS)* because it contains the knowledge of an expert, collected heuristically or by experience. KBS simulates the reasoning mechanism of a human by applying specific knowledge to the case to be accomplished. The cases to be solved generally require human intelligence. A KBS has to combine specialized knowledge with intelligence, as well as a human does while solving or deciding about a problem. The knowledge is represented as data or rules in the computer symbolically. These symbols help the system to make decisions. The knowledge can be gathered from books, manuals or a human expert. The data is converted to knowledge by using some mathematical or logical presentations, which a computer can make use of it as facts or rules of a KBS.

Some of the application areas of the KBS are medical treatment, chemistry, microbiology, engineering failure analysis, fault analysis and technological risk management systems, risk management systems, troubleshooting systems, electronics, thermodynamics, knowledge representation, climate forecasting, decision making, decision planning, chemical process controlling, education, scheduling, planning, agriculture and geographical information systems (GIS). Educational corporations are the corporations that use the KBS more frequently because curriculum planning and preparing schedules manually is a quite complex process.

Expert systems can be developed by getting use of other artificial intelligence techniques like GA, Fuzzy Logic (FL) or Neural Networks (NN), which try to help to simulate different aspects of human intelligence to computers. Thus, the usage of an ES mechanism can be integrated with a GA by using some of the components of the ES with the GA. This mechanism can be defined as a “hybrid” system to be used in optimization problems.

There exist many studies in which the GA and ES techniques are used together. The application areas that most of the hybrid studies are made are product design (Chaoan, 2007), image processing (Yu, Zhao, Ni & Zhu, 2009), material handling (Hamid, Mirhosseyni & Webb, 2009), cost management (Chou, 2009), different application areas of decision making like decision making in apparel coordination in fashion (Wong, Zeng & Au, 2009) and decision making for selecting basketball players (Ballı, Karasulu, Uğur & Korukoğlu, 2009) and different sectors and optimization problems like optimization of optical measurement systems (Otero, Sanchez, & Alcala-Fdez, 2008), composite laminate design with various rule constraints (Kim, 2007) and optimum location search (Chakravorty & Thukral, 2009).

Conventional methods use algorithms and data structures to solve a problem. For the solution of more difficult problems, heuristic strategies, which act as the human brain, are needed (Abraham, 2005). The rule based systems contain rules that help to formalize the definition of such difficult problems. It uses the rules related to the problem and evaluates or processes these rules in order to find a solution to the problem. These rules can be represented in different formats according to the needs of the system and they are recalled to solve the problem. Mostly, mathematical and logical representations are used because they are easier to integrate them to a computer program. One of the most popular and useful ways is to represent the rules as “If – Then” statements. A rule based system does not have to be an expert system; instead, different rule based mechanisms also exist. There is an obvious similarity between rule based systems and GA because a typical GA also evaluates the chromosomes according to fitness functions and which are implemented according to the rules.

1.1 The Aim of the Thesis

There are many studies, in which GA is used with other branches of artificial intelligence like expert systems, fuzzy logic or neural networks. Although ES and its components are combined with GA techniques in some studies to solve optimization problems, the rule base component of ES, isolated from ES, used within GA for solving optimization problems is considered as another research subject. Here emerges the concept of Rule Based Genetic Algorithms (RBGA). In the thesis, since rule base component of the ES is used as a part of the curriculum planning system, the system itself is not an ES; but a rule-based GA is in question.

Rule based methods are deterministic but GA indeed does not use deterministic rules and it contains randomness. It does not guarantee to converge to the solution within a fixed time (Sivanandam & Deepa, 2008). Our contribution to the literature is using the deterministic rule base component of an expert system within the fitness function of the genetic algorithm to prepare a curriculum plan for a specific course via a *rule-based genetic algorithm*. The rules are saved in the system both in logical and mathematical representation. The mathematical representation is then used to obtain the initial population of the GA. Saving the rules in these two formats (logical representation with XML and mathematical representation with matrices) brings the project flexibility and takes the advantage of adaptability of XML to any environment and representation formats.

The training data of this study is the in-service training data of a software company. There are rules among the parts of the training data. These rules are the prerequisite rules among the modules, which makes the optimization problem more difficult to manage. To ensure that the obtained results with the training data mentioned here are reliable, a control data having the same characteristics, which includes the parts of a database course, which is given in computer programming departments, is used. The thesis also contains an automated parameter tuning mechanism. With the help of the parameter tuning process, we also aimed to obtain more effective solutions to curriculum planning problem. With different parameter combinations of the GA, a set of curriculum plans are obtained for both datasets as output. These results are then evaluated with statistical analysis to find the most appropriate plan. The parameter combination giving the best curriculum plan is also discussed in respect to the values of the parameters.

The two datasets of the project differ from other datasets, which are used to solve optimization problems. They have tight prerequisite rules, which affect the size of the rule base and difficulty of the sequencing operation. This is the main reason of evaluating the module range in terms of reliability. In order to decide whether the modules' range is valid or not, it is needed to make a correlation test. In the correlation test, the output of the software is compared with the suggestion of a human expert. All of the results obtained with different parameter combinations of the GA are tested in order to find the most reliable range. The parameter combination giving the best module range is also important because that combination is considered as the best to solve this type of problems with GA. The most appropriate parameter combination giving the most reliable range is also verified with the results of the control data.

It is aimed to implement a generic GA to be used for preparing the curriculum plan for any kind of educational foundation; it can be an education plan for the courses of a faculty to put the courses in an optimum range or training material of in-service training programs in companies. The software developed for this purpose will be helpful in cases where the instructors have trouble with preparing an education program for their students / trainees. The study also has a different application area for the XML technology. The XML files include the rule base data as the input of the initial population of the genetic algorithm and the timetable output can also be saved in XML format. XML is chosen because it is a generic data format, which can be transferred and parsed by different platforms like programming languages or databases.

1.2 Organization of the Thesis Chapters

The first chapter of the thesis is considered as a welcoming about the thesis subjects emphasizing the aim of the thesis. In Chapter Two, GA, with all its mechanisms is introduced. The idea behind, and the biological terms used to define a GA is explained in detail. In Chapter Three, a detailed literature survey about GA and its usage in optimization problems takes place.

Chapter Four includes the problem definition and the sample cases used in the tests are introduced. Chapter Five is about the analysis of the problem and the solution generated is explained in detail. The sixth chapter is about the software development environment with all its cooperative technologies like database design and XML technology. Chapter Seven explains the results gathered by the execution of the system as a conclusion. A detailed analysis of the results resides in this chapter. Depending on the previous chapter, Chapter Eight includes the comments about the results of the study and suggests a future work. The MS Excel outputs, the tables including the most reliable module ranges, the source code of the software and XML files are also given in Appendices.

CHAPTER TWO
GENETIC ALGORITHMS

Genetic algorithms (GA) are introduced in 1970's by John Holland (1975). Holland is the person who had thought of simulating the Darwin's evolution theory in computer environment. Later on, his student Goldberg (1989) had developed the GA notion and thereupon, GAs became the most popular branch of evolutionary programming as known today. GAs are stochastic search algorithms which are widely used to find the optimum result as the solution of a problem in cases that the problem cannot be solved in a polynomial execution time. GA works on large populations of possible solutions instead of a single individual. This is the main point that a GA differs from other heuristic search methods. It obtains the set of best possible solutions with iterative methods as the answer of a complex problem. For this reason, it plays a great role on artificial intelligence, computation and evaluation models. Since natural selection in nature affects the biological systems on the world, evaluating the artificial systems with a similar selection mechanism is a vital component of artificial life.

Today GA is the most popular branch of evolutionary programming because the reproduction process, as the transportation mechanism of the genetic material in living organisms, is simulated to reach the best individuals of the population as occurs in natural life. GA applies some genetic operators to the individuals of its population to improve them. The improved, "better" individuals became the new members of the population instead of the older ones. The individuals who accommodate to the natural conditions survive and the ones which cannot stand to the conditions die. As the result of the genetic reproduction mechanisms, the genetic diversity of the individuals causes the algorithm to reach many different possible solutions. Since GA leads a parallel search mechanism among the possible solutions, the result of the genetic algorithm is not a simple individual, but the set of the individuals, whose properties are closest to the required properties in given conditions.

2.1 Terms of Genetic Algorithms

Since a GA is inspired from the nature, the terms used in these algorithms are taken from the biological terms. Within the cells in the living organisms, there are big molecule structures, which are called *chromosomes*. Within the chromosomes there are individual *genes*. Each gene on a chromosome encodes a specific feature of the individual (a person's eye color or height that is identified by specific genes) and the values of the genes are used to evaluate individuals.

When two individuals mate, according to the laws of sexual reproduction, both parents pass their chromosomes onto their offspring. In humans, who have 46 paired chromosomes in total, both parents pass on 23 chromosomes each to their child. The two chromosomes come together and swap genetic material, and only one of the new chromosome strands is passed to the child. In sexual reproduction, genes are exchanged among each chromosome couple and two new children chromosomes are formed. Sometimes the genes of the parents are copied and passed to the offspring as identically the same. If only a nucleotide exchange, which is the smallest unit of DNA, occurs between the parent and the offspring, it is called mutation. To bring up more qualified generations, the chromosomes with higher quality must be chosen.

Sequences of genes being chained together in chromosomes make up the DNA of an individual. According to the Pittsburgh approach (Lin & Wei, 2009), each chromosome represents a complete solution to a problem. For this reason GA tries to obtain a set of best solutions to the given problem. With this approach, the possibility to transfer the better features of a qualified population to the next generations is higher because GA produce successful solutions and successful solutions have better genetic material to transfer.

There are three more approaches (Michigan approach, Iterative Rule Learning (IRL) approach and Genetic Cooperative – Competitive Learning (GCCL) approach), which basically adopt the idea of "one chromosome contains one rule"

(Rodriguez, Escalante & Peregrin, 2011). How to represent a chromosome is tightly related with the characteristics of the problem to be solved.

The main components of the GA can be listed as follows:

- A problem to solve
- Encoding
- Initial population
- Selection of parents
- Evaluation (Fitness) value and function
- Reproduction operators
- Elitism
- Stopping criterion

To generate a GA, there must be a problem, which is not quite easy to find a solution with traditional search methodologies. Some problems may take very long time to be solved with linear methods. In such cases it is consulted to a GA solution in widely differing application areas.

2.1.1 Encoding

The input values of a possible solution are represented in a chromosome in different ways. This representation is called *chromosome encoding* and there are several different methods to handle the encoding task like binary encoding (0s and 1s), real number encoding, integer or literal permutation encoding and general data structure encoding (Kaya, 2009). The first encoding type that Holland suggested was binary string representation, where the chromosome consists of only 0s and 1s (Holland, 1975).

Permutation coded GA is used for two purposes. One is ordering, in which the elements occur before the others. The other one is adjacency, where the neighborhood between two elements has importance. In permutation coded GA, the chromosomes cannot be encoded as if they were bit strings. Instead, nonrecurring sequence of the elements on the chromosome plays a severe role on GA. Therefore some crossover methodologies mentioned in further sections were developed only for permutation coded GA.

2.1.2 Initial Population

A set of chromosomes representing a set of solutions to a specific problem is prepared before the GA is run. This set of individuals at the very beginning is called as the *initial population* of the GA. The initial population is prepared randomly, mostly generated from a single chromosome representing a sample solution for the problem. Each chromosome in the population is also called an *individual*. The number of individuals composing the initial population has an effect on the performance of GA. It directly affects the amount of genetic material which is included to the search. There is not a rule to determine the number of individuals in a population (Sivanandam & Deepa, 2008). On the contrary, it has to be chosen according to the characteristics of the problem. In the thesis the population size is in the interval of 100 – 200.

2.1.3 Selection of Parents

Through the generations of the GA, the chromosomes to be transferred to the next generation should be chosen with regard to some rules. These rules have been simulated from the Darwinian evolution theory. This theory states that the nature applies a “natural selection” mechanism on living things to find the best individuals to survive (Maulik, Bandyopadhyay & Mukhopadhyay, 2011). Better individuals can transfer better genes to next generations. The same rule is available in GA. There are many selection methods that can be applied on the chromosomes like tournament selection, roulette wheel selection and linear rank selection.

Tournament Selection: A random group of individuals are chosen from the population. The best individual in the group is chosen as done in a football championship (Teams play with each other and the best team wins) (Elmas, 2007).

Roulette Wheel Selection: The selection probabilities of the chromosomes are placed in a roulette wheel as in a pie chart of percentages and the wheel is rotated. The individual is selected according to the point that the needle in a roulette table shows. The one having the bigger percentage in the pie is more probable to be chosen.

Linear Rank Selection: The individuals are ranked according to their evaluation values. These selection methods all aim to choose more qualified chromosomes to transfer their genetic material to the next generation (Greffenstette & Baker, 1989).

In the selection mechanism, the higher probabilities of the chromosomes to be chosen has importance, but the chromosome having the higher probability may not be chosen. Randomness of the selection mechanism of GA is the most dominant factor of the evolution process.

2.1.4 Fitness Value and Fitness Function

Selection operator selects the chromosomes in the population to reproduce and bring up more quality generations according to the evaluation data of the chromosomes. Once the initial population is produced, the evolution process starts. The only information that GA needs to perform the evolution task is some measure of *fitness value* about a point in the space (sometimes known as an objective function value). This value gives information about closeness of the individual to the optimal solution (Hamid, Mirhosseyni & Webb, 2009). Once the GA knows the current measure of "goodness" about a point, it can use this to continue searching for the optimum. The *fitness* value of an organism is the surviving probability of the organism in order to reproduce. It is a measurement of how appropriate solution it encrypts. An individual having a better fitness value is more likely to be selected to produce children for the next generation. Fitness value is calculated by the help of a *fitness function*. GA deals with the problems that maximize the fitness function (Sivanandam & Deepa, 2008).

It is an important advantage of the genetic algorithms, that the chromosomes are selected and evaluated according to their fitness values, not any other criteria. Therefore GA does not require any problem – specific knowledge. The only mechanism to be programmed is the fitness function. Once the fitness function calculates the fitness values of the individuals, three kinds of fitness values should be taken into consideration. These are the best, average and worst fitness values. Best fitness value gives an idea about the performance of GA. Especially when parameter tuning is done, the same algorithms is run for different parameter combinations. In this case the best fitness values of results with different parameter combinations gives hints about the right parameter combination. Average fitness value gives an idea about the average solution and the worst value about the worst solution (Shukla, Tiwari & Kala, 2010).

2.1.5 GA Operators

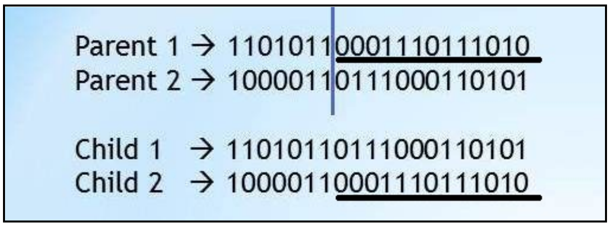
The GA is first run on the initial population and is transferred to another population by means of a kind of *operators* (methods) like reproduction, crossover or mutation. In reproduction, as stated in the elitist strategy, the selected two parents are transferred to the next generation without changing their genetic contents (Mendes, 2008). Crossover and mutation are the main operators which are applied on the selected chromosomes to obtain new offspring.

2.1.5.1 Crossover

In sexual reproduction, *crossover* occurs; genes are exchanged among each chromosome couple and two new children chromosomes are formed. There are several ways to accomplish this operation. The type of the crossover method to be applied depends on the type of chromosome encoding. The most common ones are uniform crossover, one – point crossover, two – point crossover, position – based crossover and partially – mapped crossover, which is mentioned below:

Uniform Crossover: A template chromosome composed of binary numbers (0s and 1s) in the same length with the parent chromosomes is used. Bits of the parent chromosomes are interchanges in positions where the binary template has “1” (Maulik, Bandyopadhyay & Mukhopadhyay, 2011). With uniform crossover, each gene of the chromosome has a chance to be a crossover point but it should be used for small population sizes (Picek & Golub, 2010).

One – Point Crossover: In bit string coded chromosomes, a randomly chosen point on the chromosome is selected for both of the parents chosen to mate. The two parents exchange their genetic material with each other from the selected point of the chromosome (Shukla, Tiwari & Kala, 2010). This point is called the *crossover point* or the *cut point*. As a result of this operation, the first offspring takes the first part from Parent 1 and the second part (after the point chosen randomly) from Parent 2. The same applies for the second chromosome, first part from Parent 2 and the second part from Parent 1 (Coley, 1998), as depicted in Figure 2.1.



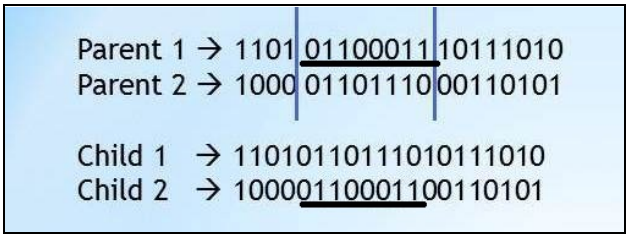
```
Parent 1 → 1101011|0001110111010
Parent 2 → 1000011|0111000110101

Child 1  → 11010110111000110101
Child 2  → 10000110001110111010
```

Figure 2.1 Example for 1 – point crossover: The offspring are produced by exchanging the two parts of a chromosome divided from the crossover point shown with a bar.

Two – Point Crossover: In bit strings, the genetic materials of the parents between randomly chosen two crossover points are exchanged with each other to produce two new individuals (Figure 2.2). This kind of crossover helps the genetic diversity of the population (Shukla, Tiwari & Kala, 2010). Two – point crossover is generally considered better than one – point crossover (Sivanandam & Deepa, 2008).

One – point and two – point crossover operators work properly for the chromosomes encoded as the bit strings but with the chromosomes encoded with permutation encoding (ordered chromosomes), it does not work properly. In permutation encoding, the genes of the chromosome are not allowed to repeat in the chromosome. For this reason some unwanted offspring may be produced with standard one – point and two – point crossover. To avoid this problem, another crossover technique is developed. Order crossover is used in such cases.



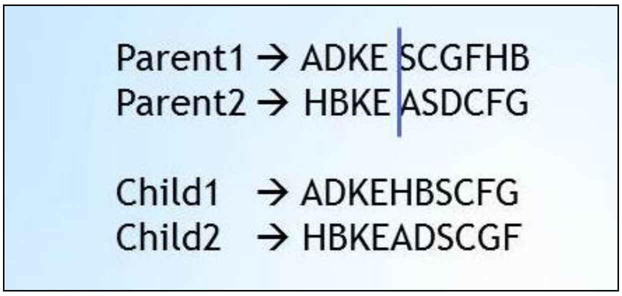
Parent 1 → 1101 01100011 10111010
Parent 2 → 1000 01101110 00110101

Child 1 → 11010110111010111010
Child 2 → 10000110001100110101

Figure 2.2 Example for 2 – point crossover: The interval between the two crossover points is exchanged from Parent 1 to Child 2 and from Parent 2 to Child 1.

One – Point Order Crossover: In this type of crossover, the chromosome up to the crossover point is taken directly from the parents; the rest of the chromosome is completed with the genes in the same order with that of the parent's (Davis, 1991) as shown in Figure 2.3.

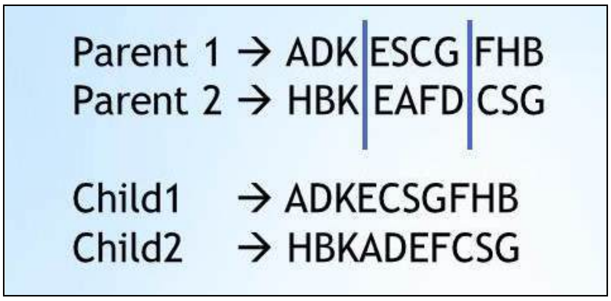
Two – Point Order Crossover: In permutation coded chromosomes, two crossover points are determined and the first and the last parts of the parents are transferred directly to the children. This means, Child 1 inherits the first and last parts of Parent 1 and Child 2 inherits the first and last parts of Parent 2 directly. But the middle section of Child 1 is taken from the unused genes of Parent 2 and middle section of Child 2 is taken from the unused genes of Parent 1 in the order they appear in the chromosome as explained in Figure 2.4.



Parent1 → ADKE SCGFHB
Parent2 → HBKE ASDCFG

Child1 → ADKEHBSCFG
Child2 → HBKEADSCGF

Figure 2.3 Example for 1 – point order crossover: The first part of the Child 1 is taken from Parent 1 and the second part of Child 1 is taken from Parent 2, the genes which are not taken from Parent 1.



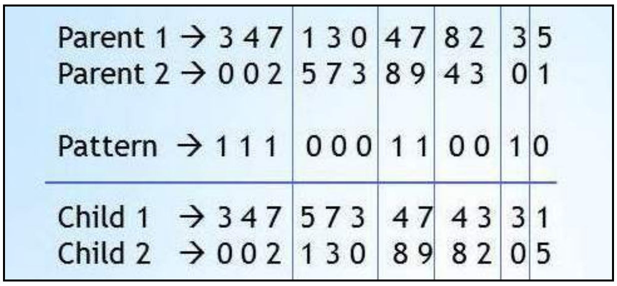
Parent 1 → ADK|ESCG|FHB
Parent 2 → HBK|EAFD|CSG

Child1 → ADKECSGFHB
Child2 → HBKADEFCSG

Figure 2.4 Example for 2 – point order crossover: The interval between the two crossover points is exchanged from Parent 1 to Child 2 and from Parent 2 to Child 1.

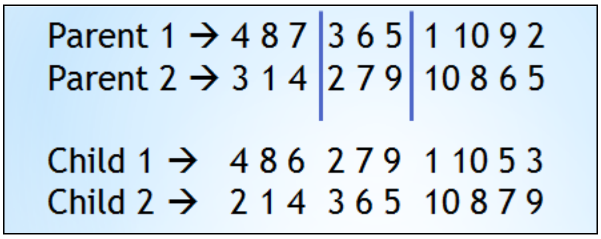
Position Based Crossover: According to a given pattern, the parents exchange their genetic material. An example to position based crossover is given in Figure 2.5. The genetic material corresponding to the 0s in the pattern is exchanged in the example.

Partially Matched Crossover (PMX): Two crossover points are selected randomly as in two – point crossover. The genetic material of parents is divided into three sections with the crossover points. The middle sections of the parents are exchanged, but since this operator is applied on permutation encoded chromosomes, repeating genes must be avoided. To solve this problem, a repair operator is used (Sivanandam & Deepa, 2008). While the middle section of Parent 2 is inserted in the middle section of Parent 1, the original genes in the middle section of Parent 1 goes to the positions of Parent 1, where resides the genes from Parent 2. As shown in Figure 2.6, when the genes 2, 7 and 9 from Parent 2 are transferred to Parent 1, the genes 3, 6, 5 of Parent 1 goes to the places of 2, 7, 9 in Parent 1 to form Child 1. Same applies for Child 2 when 3, 6, 5 are transferred from Parent 1 to Parent 2.



Parent 1 →	3 4 7	1 3 0	4 7	8 2	3	5
Parent 2 →	0 0 2	5 7 3	8 9	4 3	0	1
Pattern →	1 1 1	0 0 0	1 1	0 0	1	0
Child 1 →	3 4 7	5 7 3	4 7	4 3	3	1
Child 2 →	0 0 2	1 3 0	8 9	8 2	0	5

Figure 2.5 Example for position – based crossover: The genes are exchanged according to the pattern. The pattern is decided randomly also.



Parent 1 →	4 8 7	3 6 5	1 10 9 2
Parent 2 →	3 1 4	2 7 9	10 8 6 5
Child 1 →	4 8 6	2 7 9	1 10 5 3
Child 2 →	2 1 4	3 6 5	10 8 7 9

Figure 2.6 Example for partially – matched crossover: The genes between the chosen interval are exchanged as in 2 – point order crossover and a repair operator is used to avoid recurrence.

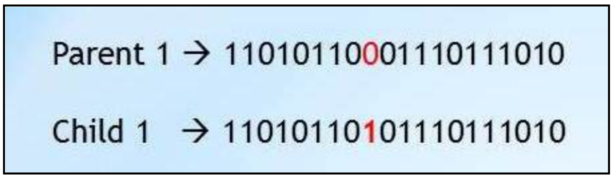
2.1.5.2 Mutation

The crossover in GA is controlled with a probability value. If the crossover probability is high, most of the chromosomes are put to the crossover operation. But sometimes the genes of the parents are copied and passed to the offspring without crossover, as identically the same. If only a gene is changed from parent to the child, then it is called *mutation*. This method avoids the local minimum and supports genetic diversity.

By applying mutation on a population with a reasonable mutation rate, the algorithm may be able to find better solutions among mutated chromosomes. There are several ways to apply mutation on chromosomes. Some frequently used types of mutation are uniform mutation swap mutation, inversion mutation and insertion mutation.

Uniform Mutation: In bit strings, mutation is simply the process of changing the value of a randomly chosen gene (0, if it is 1, 1, if it is 0) (Shukla, Tiwari & Kala,

2010) as given in Figure 2.7.

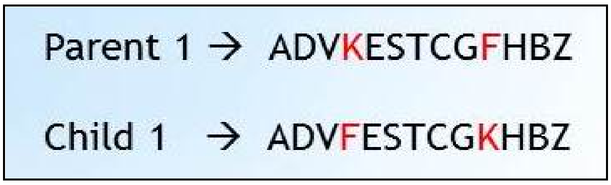


Parent 1 → 11010110001110111010

Child 1 → 11010110101110111010

Figure 2.7 Example for uniform mutation: The gene to be mutated is chosen randomly.

Swap mutation: In this type of mutation, two randomly chosen genes are swapped (Chiou & Wu, 2009). It can be used in both bit string and permutation coded chromosome representations (Figure 2.8).



Parent 1 → ADVKESTCGFHBZ

Child 1 → ADVFESTCGKHBZ

Figure 2.8 Example for swap mutation: Two randomly chosen genes swap.

Inversion Mutation: A random interval is determined on the chromosome and the genes in this interval are reversed to produce two offspring different than their parents (Figure 2.9) (Kaya, 2009), (Molla-Alizadeh-Zavardehi, Hajiaghaei-Keshteli & Tavakkoli-Moghaddam, 2011).

Insertion Mutation: A randomly chosen gene is inserted to a randomly chosen position on the chromosome. If the position to be inserted is located before the original location of the gene, the genes from the insertion position are shifted one position to the right. But if the position to be inserted is located after the original location of the gene, the genes from the insertion position are shifted one position to the left (Meng, Zhang & Li, 2010) as shown in (Figure 2.10).

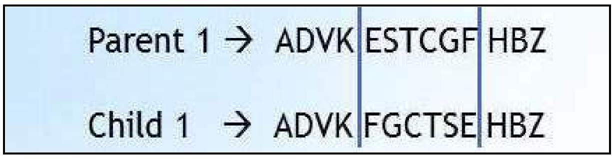


Figure 2.9 Example for inversion mutation: The genes between the two randomly chosen points of Parent 1 are reversed to produce the Child 1.

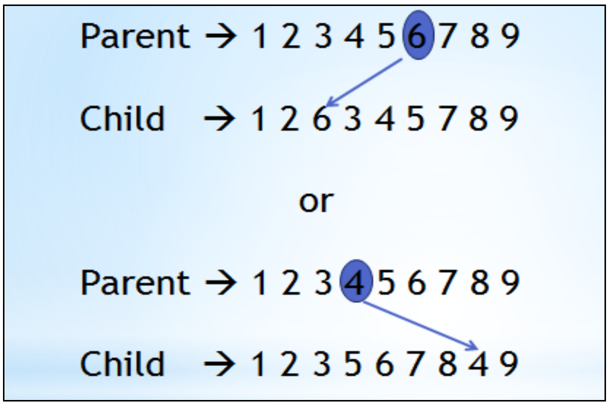

Parent → 1 2 3 4 5 6 7 8 9
Child → 1 2 6 3 4 5 7 8 9
or
Parent → 1 2 3 4 5 6 7 8 9
Child → 1 2 3 5 6 7 8 4 9

Figure 2.10 Example for insertion mutation: The chosen gene changes its place while shifting the other genes to left or to right.

2.1.6 Elitism

When the new generation of individuals is generated, some individuals having the best fitness values may not be selected for the reproduction process. In order to prevent the loss of the best individuals, *elitism* mechanism is applied to the population. That is, some of the best chromosomes of the previous generation are copied to the new population directly, without applying any genetic operator. Other individuals are selected and reproduced for the next generation in a classical GA process (Maulik, Bandyopadhyay & Mukhopadhyay, 2011). This mechanism protects the best individuals against crossover or mutation.

Elitism is a powerful strategy improving a GA’s performance in a positive way (Sivanandam & Deepa, 2008). Generally there are two basic methods to apply elitist strategy to a population (Deb, 2001). The first one is to copy directly some percent of the population directly to the next generation. The second way is to compare two offsprings with their parents and choose the better two individual among the four for the next generation (Mokhtari, Abadi & Zegordi, 2011). In both cases, elitism should be applied with a reasonable amount of individuals. Transferring all best individuals of a population directly to the next generation may cause lack of diversity. Not applying elitism may also cause to lose best individuals.

2.1.7 Stopping Criterion

In a typical GA, an initial population of individuals is generated randomly. Each step of the iteration is called a *generation*. The individuals in the current population are evaluated according to the criteria, which was defined before the iterations start. These criteria are defined by the fitness function of the algorithm. To form a new population as the next generation of the algorithm, individuals are selected according to their fitness values. By doing so, the expected number of times an individual is chosen is approximately proportional to its relative performance in the population.

The number of generations is a common *stopping criterion* for the GA. The algorithm has to stop somewhere and at the end, must have the set of best results. There are several ways to stop the GA:

- A certain number of generations can be assigned to stop the program
- The program may stop when there occur no changes in the fitness values of the individuals (if the solution set does not improve)
- Fitness value reaches its maximum (Srndic, Pandzo, Dervisevic & Konjicija, 2009).

Since it has some disadvantages to use a standard GA, researchers try to find the best GA to solve the optimization problems in the best way it can. Traditional GA highly depends on the initial population and tends to converge rapidly. The genetic operators may also decrease the diversity of the individuals in the population. As a result of these handicaps, many studies are done to handle the problems of GA.

2.2 The Steps of a Standard GA

The following pseudocode can be written for a standard genetic algorithm:

```
initpop P
For each $solution_i$ from P
    calculateFitness($solution_i$)
repeat
    select parents $solution_1$ and $solution_2$ from P
    child = crossover($solution_1$, $solution_2$)
    mutate(child)
    calculateFitness(child)
    replaceChild(P, child)
until stoppingcriteron
```

2.3 Application Areas of Genetic Algorithms

GA can be used in a wide scale of applications in control systems engineering, materials engineering and electrical engineering. These applications include topics like:

- Speech recognition and natural language processing (NLP),
- Telecommunication and network design,
- Optimization,
- Economics,
- Scheduling in different application areas,
- Automatic programming and machine learning,
- Computer – aided design (CAD),
- Game theory,
- Astronomy and weather forecasting,
- Mathematics,
- Chemistry and biology,
- Bioinformatics and ecological models,
- Data mining.

To find solutions of the problems of these areas, GA can be combined with other AI techniques like Robotics, Fuzzy Logic (FL), Neural Networks (NN) or Machine

Learning. Among the application areas, solving the optimization problems of the systems is the most popular one. It is an iterative procedure that consists of a constant-size population of individuals, each one represented by a finite string of symbols, encoding a possible solution in a given problem space. It is called the search space, which comprises all possible solutions to the problem.

CHAPTER THREE
LITERATURE OVERVIEW

GA is a quite popular research area in computer science and there are many studies including different aspects of GA. The studies including GA can be classified into two main groups. Some studies deals with the performance of the GA, where some combine GA with other artificial intelligence techniques. Below some studies of both groups are listed.

3.1 Optimization

GA is mostly used as an optimization technique. For this reason, many of the studies using GA deal with some optimization problems. There are several optimization types that GA is used like global optimization, constrained optimization, combinatorial optimization and multi-objective optimization (Lau, Tang, Ho & Chan, 2009), (Kaya, 2010). For example since risk management has become one of the most studied topics with GA, a heuristic approach to portfolio optimization problem in different risk measures is handled by using this methodology (Chang, Yang & Chang 2009).

There are many studies mentioning the solutions of optimization problems with Rule-Based GAs (RBGA) because rule based systems play an important role to improve the performance of the search methodologies. In the usage of rule based systems with GA, rule base may help GA while evaluating the individuals of the new generation (Wang, Liu & Yu, 2009), (Choy, Leung, Chow, Poon, Kwong, Ho & et al., 2011) or GA can be used in rule extraction. A GA can be based on some heuristic rules for problems of large size (He & Hui, 2008), (Fernandez, del Jesus & Herrera, 2009). Except optimization problems, rule based systems are also used in genetic programming (Weise, Zapf & Geihs, 2007), network security (Mishra, Jhapate & Kumar, 2009), scheduling (Zhang & Tu, 2010).

Another topic that needs an optimization solution is feature selection. Selecting the optimal set of features among many of them is done by using a GA (Li, Zhang & Zeng, 2009). Like feature selection, decision making is a remarkable application area for the evolutionary techniques because solutions to such problems can be obtained effectively with genetic algorithms with lower costs of processing times. Order – acceptance problem with tardiness penalties is a good example of this kind of problems (Rom & Slotnick, 2009). In molecular biology domain, multiple sequence alignment issue plays an important role and an approach different than GA like Decomposition with GA (DGA) is applied. The overall performance of DGA has been found better than traditional GA (Naznin, Sarker & Essam, 2010). Machining sequencing is one of the application areas of GA, in which special chromosome structures and encoding schemes can be applied according to the problem definition (Shu, Gong & Wang, 2010).

3.2 Scheduling and Timetabling

Scheduling and planning problems can be considered as the optimization problems because researchers seek for the optimal solutions to solve this kind of problems. While seeking the optimal solution to scheduling problems, the value of population size, the design of the fitness function, and parameters of genetic operators should be decided carefully (Lee, Wu & Liu, 2009). Route planning problem is one of those in which GA is used (Wu, Shih & Chen, 2009). In the study, an efficient solution to a cross-fab route planning problem for semiconductor wafer manufacturing is handled and quite satisfactory results are obtained by implementing a standard GA with one-point crossover operator. In manufacturing environment, GA can also be used for scheduling a decision support model to minimize job tardiness (Choy, Leung, Chow, Poon, Kwong, Ho, et al., 2011).

Scheduling problems also arise in multiprocessors and parallel and distributed systems. Studies done so far on these application areas with GA have shown that Artificial Immune systems, especially Immune GA (IGA), perform well in reducing the number of iterations and exploring the search space to find the solution

(Moghaddam & Monyadi, 2011). In production scheduling problems, GA is used and can be combined with different mathematical models to solve the problem with better results (Fakhrzad & Zare, 2009).

Using GA is a popular technique to solve job – shop scheduling problems. These techniques can easily been applied to any kind of job – shop problems like no – wait and blocking job – shops (Brizuela, Zhao & Sannomiya, 2001). Combining GA with other local search techniques ends up with more effective results. A study has been done on job – shop scheduling problem, where it brings assertive results thanks to the crossover technique used in the hybrid GA (Tseng & Lin, 2010). Researchers have shown that dividing the problem into sub problems and performing a hybrid GA on these parts improves the solution quality on job – shop scheduling problems (Pan & Huang, 2009). Another study have shown that the results of improved adaptive genetic algorithm (IAGA) to a job – shop scheduling problem reports a more efficient production and more efficient usage of the machines (Wang & Tang, 2011). Simulated annealing is also another method for job – shop planning and scheduling problem. In one of the studies it has been combined with GA as Adaptive Annealing GA (AAGA) to solve the local convergence problems of a classical GA and improving the convergence rapidity of GA (Liu, Sun, Yan & Kang, 2011).

Using GA methodologies in multi – product parallel machines help to reduce the setup time for sheet metal shops and the same job can be routed in multiple machines with a reduced make – span (Chan, Choy & Bibhushan, 2011). The way of representing the chromosomes also affects the performance of the scheduling process in multi – product systems. (Ramteke & Srinivasan, 2011). For scheduling simultaneous multiple resources, bi – vector encoding GA (bvGA) is applied as another solution method. In this method, chromosome representation of GA and rules for resource assignment play an important role in solving the problem. bvGA improves the solution quality and reduces the computation time as well (Wu, Hao, Chien & Gen, 2011). GA basically offers efficient solution techniques with minimum number of GA variables in scheduling problems and low computational burden (Sasikala & Ramaswamy, 2010).

Arrival Sequencing and Scheduling (ASS) is also an important application area for evolutionary approaches. Especially Ant Colony Systems (ACS) seems to be an effective way to solve such kind of traffic control problems. The experimental work on ACS for ASS outperforms well and reduces the computational burden in optimization (Zhan, Zhang & Gong, 2009). ASS can be solved with Bee Evolutionary Genetic Algorithms (BEGA) and this approach helps to obtain an optimum landing sequence and landing time effectively (Wang, 2009). An aircraft category based GA is used in a study which obtains better results in a real time application (Meng Zhang & Li, 2010).

Similarly, aircraft landing scheduling problem is considered as a tough optimization problem with many hard constraints since it has to be handled in real time. As distinct from the traditional optimization methods, researchers have obtained better solutions by using genetic algorithms (Yu, Cao, Hu, Du & Zhang, 2009). Different GA methodologies have also been applied and compared in aircraft Departure Sequencing Problem (DSP) like Basic GA, Adaptive GA and Improved GA (IGA). Among these methodologies it is concluded that IGA has a better performance when compared to Basic and Adaptive GA methodologies (Wang, Hu & Gong, 2009). Ripple Spreading GA (RSGA) is one of the techniques applied on aircraft sequencing problems, which inspires from the ripple – spreading phenomenon of nature in liquid surfaces. This methodology has many advantages like being flexible, extendible, memory – efficient and filtering the bad solutions automatically (Hu & DiPaolo, 2011). In some of the solutions found for airline rostering problems, novel chromosome representation techniques are introduced, improved crossover and mutation operators are applied and both operators can be used alternatively (Souai & Teghem, 2009).

Nurse scheduling problem is very popular research area and GA is used to prepare an optimal schedule taking the constraints of the job into consideration (Tsai & Li, 2009). Planning surgical operations require an effective scheduling to prevent any violations in human resources and conflicts in operating rooms. GA solves the scheduling problem of surgical activities in terms of time and resource constraints (Roland, Di Martinelly, Riane & Pochet, 2010). Other search techniques like Tabu Search can also be combined with genetic algorithms to solve complex scheduling problems like scheduling an in-line-stepper in a semiconductor fab (Chiou & Wu, 2009) or compressor selection in natural gas pipelines (Nguyen, Uraikul, Chan & Tontiwachwuthikul, 2008). Hybrid GA methodologies are also applied to solve no-wait job shop scheduling problems (Mokhtari, Abadi & Zegordi, 2011). A hybrid system may contain local search mechanism and a traditional GA. Local search, in this case, is used to improve the initial population (Whitley, 1995). Using multi – objective evolutionary algorithms (MOEA) in scheduling problems has become a popular problem solving technique. With this approach, researchers have reached well – performing results. Multi – objectivization concept has developed and has been supported with helper objectives to find an optimum sequence of the objectives (Lochtefeld & Ciarallo, 2010).

In education domain, GA is also used to prepare timetables and schedules. There exist so many studies to develop different scheduling methods for educational timetabling problems. Timetabling problems are considered as NP-hard problems and most of the studies have dealt with educational timetabling by constructing some methodologies to achieve timetabling task for an educational issue (Burke, McCollum, Meisels, Petrovic & Qu, 2007), (Aldasht, Alsaheb, Adi & Qopita, 2009) (Khonggamnerd & Innet, 2009) (Raghavjee & Pillay, 2010).

Researchers have looked for alternative solution approaches to the distinct branches of timetabling problems like examination timetabling (Carter & Laporte, 1996), (Derakhshi & Zandi, 2010), (Pillay & Banzhaf, 2010), (Cupic, Golub & Jakobovic, 2009), course timetabling (Carter & Laporte, 1998), (Abdullah, Turabieh, McCollum & McMullan, 2010a), (Abdullah, Turabieh, McCollum & McMullan, 2010b), (Chinnasri & Sureerattanan, 2010), (Jat & Yang, 2011) , (Ayob & Jaradat, 2009). Some researchers have tried to classify (Bardadym, 1996) and automatize the timetabling problems (Burke, Jackson, Kingston & Weare, 1997), (Schaerf, 1999), (Burke & Petrovic, 2002).

On the other hand, university timetabling became another type of timetabling problem, in which many remarkable studies have been done. The hard constraints and soft constraints of a timetabling problem and detecting these constraints precisely play a great role in finding the most appropriate timetables (Petrovic & Burke, 2004). Alsmadi, Abo-Hammour, Abu-Al-Nadi & Algsoon tried to solve a university timetabling problem by developing a GA to handle the constraints, diminishing the hard constraint violations (2011). Parallelization of GA is another choice to handle university timetabling problems, solving the problem with a master – slave architecture (Karol, Tomasz & Henryk, 2006). In one of the studies done on university timetabling, a hybrid grouping GA is developed and applied on a real application. It is concluded that a hybrid GA method can assign the students to the laboratory groups with a maximum capacity and less conflict (Agustin – Blas, Salcedo – Sanz, Ortiz – Garcia, Portilla – Figueras & Perez – Bellido, 2009).

Except educational timetabling, some other application areas of timetabling can be mentioned like nurse rostering (Cheang, Li, Lim & Rodrigues, 2003), (Burke, De Causmaecker, Berghe & Van Landeghem, 2004), sports timetabling (Easton, Nemhauser & Trick, 2004), transportation timetabling (Kwan, 2004), finding the best match problem among many candidates and tasks (Altay, Kayakutlu & Topcu, 2010) and grid scheduling (Adamuthe & Bichkar, 2011). Train sequencing on the railways has also been considered as a transportation timetabling problem to be solved with genetic algorithms (Chung, Oh & Choi, 2009).

Curriculum sequencing, which can be defined as a Constraint Satisfaction Problem (CSP), is one of the favorite research areas that optimization techniques like GA are used (Hong, Chen, C.-M., Chang & Chen, S.-C., 2007), (De Marcos, Barchino, Martinez, Gutierrez & Hilera, 2008) (Olsen, 2009). Even complex sequencing scenarios can also be processed by applying a model of permutation constraint satisfaction problem (De Marcos, Martinez, Gutierrez, Barchino & Gutierrez, 2008). For arranging employee training programs, GA is also preferred as a scheduling methodology and an optimal curriculum arrangement can be done easily and effectively (Juang, Lin & Kao, 2007).

For solving the curriculum sequencing problems, one approach is to develop agents by using evolutionary computation methods (De Marcos, Barchino & Martinez, 2008). Another approach is considered as the permutation – based genetic algorithms, which is used to perform sequencing optimization (Li-li & Ding-wei, 2008). Permutation – coded genetic algorithms can be applied to different problems like weapon – target assignment problem (Julstrom, 2009).

In precedence – constrained sequencing problems (PSCP), optimization is done to locate the optimal sequence with the shortest travelling time. Some hybrid genetic algorithm (HGA) techniques with adaptive local search help to produce the most effective results when compared with the results of other traditional methodologies (Yun, Gen & Moon, 2010).

One of the most famous sequencing problems is Travelling Salesman Problem (TSP). The solution to the problem aims to find the shortest path for the salesman to traverse different cities, stopping by the same city only once. Many researchers have constructed many solution methods on TSPs (Singh & Baghel, 2009). GA brings some effective solutions to TSP and some hybrid algorithms are implemented (Pop & Iordache, 2011). When GA is the point in question, diversity control is an important notion in TSP problems because if the diversity reduce rapidly, the solution to the TSP can be worse in quality. Researchers have studied on diversity control in TSP problems and gathered encouraging results (Chang, Huang & Ting, 2010).

The techniques used to find solutions to TSP problems are not widely different than the methodologies used in permutation sequencing problems. For this reason, some TSP solution methodologies can be adapted to GA. TSP is a good area of applying and testing the performances of new crossover (Deep & Mebrahtu, 2011), (Ahmed, 2010) or mutation (Kaya, 2010) operators. New type of GA, a whole with its chromosome representation and reproduction operators, can also be applied and tested on TSP. For example, a grouping based GA has been applied on multiple travelling salesperson problems (Singh & Baghel, 2009) and the results are rather promising.

3.3 GA Performance and Parameter Tuning

Except the studies combining different artificial intelligence techniques, there are other studies which only deal with improving the performance of the GA. Performance evaluations in a GA can be done in several ways.

The parameter tuning notion involves with choosing the exact control parameters for GA to run and produce the best results. The parameters, to a large extent, affect the performance of GA. The determination of the correct combination of parameters itself is also an optimization problem. The researchers have made many studies on parameter tuning. To handle the parameter control problem in GA, many solutions have been suggested. One of these even offers a scripting language implementation for controlling the parameters of evolutionary algorithms (Liu, Mernik & Bryant, 2004).

As an example of this research area, different crossover operators can be implemented (Marano, 2011) and be compared with each other (Lin & Wei, 2009), (Vazquez-Rodriguez & Petrovic, 2010). There are some studies in which the crossover techniques are surveyed for different GA types (Tutkun, 2009). But mostly, improvement in the performance of the GA is handled by making some parameter tuning operations. In order to make parameter tuning, the same GA must be executed many times with different number of generations, crossover and mutation rates and population sizes. In most of the studies, at least two or three different values are tested for each parameter and the execution times are calculated (Rau & Cho, 2009), (Choy & et al., 2011). The best parameter combination is found according to the least execution time value. But in some of the studies dealing with parameter tuning, new chromosome encoding techniques and crossover operators can be introduced (Barrero, Gonzales – Pardo, Camacho & R – Moreno, 2010).

The parameters used in a GA directly affect the performance and the quality of the result set. The analysis of the parameters itself is also a difficult task to manage. There are many studies dealing with this notion. For example, Eiben and Smit worked on the analysis of the parameters in evolutionary algorithms (Eiben & Smit, 2010). In their study, they classify the parameters into two groups: qualitative and quantitative parameters. In their study, they aimed to prove that parameter tuning is a must and it must be available in all studies dealing with evolutionary algorithms.

The parameters of a GA are meaningful when they are combined in a harmony to find solutions to the optimization problems. That is, all parameters values about crossover, mutation, number of individuals, number of generations, the selection mechanisms and elitism, must be thought together. There are several studies in which these parameters are stated as a whole (Alsmadi, Abo-Hammour, Abu-Al-Nadi & Algsoon, 2011). When the same parameter combination is used for another application area, the results can be compared and the parameter combination may give an idea about the performance of the system. Contrary, in one of the studies, instead of genetic operators such as crossover and mutation, the concept of compact Genetic Algorithms (cGA) come up with the idea of probability vector (Lee, Kim & Lee, 2011).

Elitism is one of the quite powerful weapons of GA to affect the performance of the algorithm. For this reason, it is used in many studies dealing with GAs. Especially in timetabling problems, keeping a good individual for generations is important to obtain the optimum solution. In a study solving the timetabling problem with GA, Jorge, Martin and Hector applied elitism to 1 % of the population (Jorge, Martin & Hector, 2010). They also worked on the parameter tuning strategies to improve the performance of their system about solving the academic timetabling problem (Jorge, Martin & Hugo, 2010).

Castelli, Manzoni and Vanneschi have found a novel way to transfer the genetic material of some of the good individuals from one generation to the next one. They proposed a mechanism to replace the worst individuals of the new generation with the good from the previous generation. By doing so, they gave the better individuals chance to mate and reproduce (Castelli, Manzoni & Vanneschi, 2011).

Cheng, Shi, Yin and Li have used elitist strategy for streaming pattern discovery in wireless sensor networks as a different application area of GA (Cheng, Shi, Yin & Li, 2011). The algorithm with the elitist strategy has the power to reduce the reconstruction error and is fully applicable in the wireless sensor networks area.

Dynamic Optimization Problems (DOP) is a good application area for performing the elitist strategy. Lee and McKay have studied on three well-known optimization problems and observed the behavior of the evolutionary algorithms having elitism mechanism (Lee & McKay, 2011).

As the characteristic of the fitness function of a GA, it has to produce fitness values to be accepted as “valid” for the solution of the problem. These values may tend to converge either to a maximum or to a minimum value. Maximizing the fitness values to find the optimum population of individuals is often used in GA applications (Ahmet & Zhoujun, 2010), (Li, Lv, Mei & Xu, 2010). In most of the GAs, a penalty score is calculated for the genes in the chromosome. The chromosome having the least penalty score is accepted as the best individual of the population. If the penalty score of the chromosome is written as the denominator of the fitness function, then the individual having the least penalty score has the largest fitness value. This is one of the ways of maximizing the fitness values. It can be used with optimization of timetables because the constraints of the problem can be considered as the penalty score of the individuals. The same approach can also be used in fixed – charge transportation problems (Molla-Alizadeh-Zavardehi, Hajiaghaei-Keshteli & Tavakkoli-Moghaddam, 2011) or network broadcast control of the GSM systems (Biroğul, Elmas, Çetin, 2011).

According to the problems and the solution techniques needed for the problems, GAs have also improved since they were first stated. Today, some hybrid or modified GAs are used in solving some of the problems. Marano has a study on such modified GAs, in which he has improved the GA operators also (2011). Adaptive GA (AGA) is popular in parameter optimization and finds itself some real – world application areas like optimization of traffic in a computer network (Prieto & Perez, 2008), (Fernandez – Prieto, Canada – Bago, Gadeo – Martos & Velasco, 2011).

3.4 GA and Correlation Tests

When an optimum result for an optimization problem is gathered by using GAs, the results must be tested and evaluated whether they are "*reliable*" and "*valid*" or not. There are many evaluation techniques to accomplish this task. In sequencing problems, usually the results are compared with the suggestions of a human expert. To make a reliable decision, non-parametrical tests can be applied to the results. Spearman Correlation Test is one of these test techniques. It is used when the range of the observations, not the values, is important (Sheskin, 2000). Since GAs deal with optimization and sequencing problems, Spearman correlation test has also been used with some of the studies.

Ebrahimipour and Nagasaka have used Spearman correlation to verify the results of a GA in their study about power sector performance (Ebrahimipour & Nagasaka, 2003). The result of the GA is compared with the results of Principal Component Analysis (PCA) model and Numerical Taxonomy (NT) by using Spearman Correlation test.

GA and Spearman can also be used together to make hardware performance prediction of a system (Hoste, Phansalkar & Eeckhout, 2006). The results obtained from Normalization, GA and PCA are compared and it is observed that the estimation results of GA have a better correlation than the other methods.

GA is used with Variable Subset Selection (VSS) approach in Pavan, Mauri and Todeschini's study and the models based on VSS are evaluated with Spearman rank index (Pavan, Mauri & Todeschini, 2004). For the estimation of surface-based duct parameters from radar clutter, Xiao-Feng, Si-Xun and Zheng have used Spearman rank correlation with GAs (2010). In another study, Spearman is used to evaluate the results of a GA making an optimization for the isolation of the translational efficiency bias and the traditionally generated results (Raiford, Krane, Doom & Raymer, 2011). According to their Spearman evaluations, GA, as a search based method, has performed better than the traditional method.

GA is used for the optimization of a culture medium for producing the toxic marine dinoflagellate microalga. The evaluations of the results of GA are again done with Spearman rank correlation and it is concluded that the medium generated with a GA optimization gave a better result than those of the other control media (Camacho, Rodriguez, Miron, Christi & Grima, 2011).

3.5 GA with Other AI Techniques

GA is best combined with Fuzzy Logic (FL) and Neural Networks (NN) in many of the studies. In one of the studies, fuzzy GA approach, which integrates fuzzy rule sets and their membership function sets in a chromosome, is developed to solve an optimization problem (Lau, et. al. 2009), (Fernandez, del Jesus & Herrera, 2009). Fuzzy systems can be combined with GA in a harmony to solve real – world problems like stock markets and the combination of the two methods ends up with an improved performance (Hung, 2009). Also determining the optimum weights on each edge of network traffic is an important problem to be solved to manage unbalanced networks. A study about wireless sensor networks in done by integrating FL and GA (Yun, Lee, Chung, Kim, & Kim, 2009). In the study, edge weights in the network are modeled with FL and optimized with GA.

Ant colony systems – like particle swarm optimizations (PSO) – can also find answers to scheduling problems. GA finds itself a valid application area in combining PSO with GA (Valdez, Melin & Castillo, 2011), (Anghinolfi, Montemanni, Paolucci & Gambardella, 2011). For a flow shop scheduling problem, Yağmahan and Yenisey have obtained better results than that of the other methods compared (Yağmahan & Yenisey, 2010).

GA is used with data mining in some rule extracting applications (Rodriguez, Escalante & Peregrin, 2011). They used GA for the rule extraction for a system that states a new method for data distribution in computer networks. The algorithm applied has some advantages when compared with other distribution algorithms.

The recent studies show that there are some Expert Systems (ES) which are combined with GA techniques. These techniques are mostly used in education and many scheduling systems are produced for this purpose. In some studies, rule extraction of the ES is managed by using GA (Fan, Tseng, Chern & Huang, 2009). In the study, the rules that ES uses are generated and updated in each generation. GA and ES are used together even in an unmanned aerial vehicle (UAV) navigation system (Kuroki, Young & Haupt, 2010). GA is used within a rule- based system to tune the variables of the system until the output matches the observations.

CHAPTER FOUR
PROBLEM DEFINITION

4.1 Characteristics of Data

The system is designed to find an optimum sequence for the contents of a course / training, where there are some obligatory rules among the parts of the course. This system is appropriate for the cases that not all of the course material is taken into consideration but instead, some selected parts, which are called *modules*, of the education material is in question to be placed in the curriculum plan. This means that the instructor can make a selection among the modules and make a decision about the length of the dataset. Instructor can select the modules to be included to the course via the user interface of the software as explained in Chapter Six. The information about the course material (the modules and their features) can also be added and dropped via a user interface. When modules are added for a training program, the names and module numbers are inserted in a database.

Two different data sets with same kind of properties are used in the thesis. One of them is the training data including the training program of an in – service training program of a software company. The other one is the control data set including the topics of a database course given in a computer programming department of a vocational school as the modules of the course. The names of the modules and the prerequisite module numbers for each module are listed in Appendix A for both the training data and the control data sets.

The most important criterion about the modules is the *prerequisite* conditions. When the module *i* has the module *j* as its prerequisite, this means that, to get module i, the trainee should pass the module j successfully. This can be considered as a precedence constraint which Mendes mentions in his study (2008). For the system to work properly, the time periods of all modules and prerequisite modules of the modules (if any) must be determined clearly.

The data containing the prerequisite conditions of the modules form the rule base of the system and sequencing process is based on the prerequisite rules of the modules. For this reason the rules must be transmitted to the system carefully. The rule based of the system behaves as if it was an ES component because the prerequisite modules of a module are represented as logically like **IF-THEN** statements as used in ES methodologies. An example to the **IF-THEN** statements used for the prerequisite rules of the modules is represented in Figure 4.1. For example "**If M1 and M2 then M4**" means module M4 has M1 and M2 as its prerequisites. The trainee has to be successful from modules M1 and M2 to have the module M4. A more detailed versions of the **IF-THEN** rules used for the training and control datasets are given in Appendix B.

There is an important point about the rules of the modules that there must not be any cyclic rule definition. That is, if M_p is defined as the prerequisite of M_q, then M_q must not be defined among the prerequisites of M_p.

```
IF M1 THEN M2
IF M1 THEN M3
IF M1 AND M2 AND M3 THEN M4
IF M1 AND M3 AND M6 THEN M7
IF M1 AND M2 AND M4 AND M5 AND M8 THEN M9
…
IF M1 AND M14 THEN M16
IF M1 AND M2 AND M3 AND M12 THEN M20
IF M1 AND M2 AND M12 AND M15 AND M17 AND M20 THEN M22
…
IF M1 AND M3 AND M7 AND M27 THEN M28
IF M1 AND M3 AND M7 AND M27 AND M28 THEN M29
…
```

Figure 4.1 Prerequisite rules among the modules

4.2 Obtaining the Module Features

The modules that will be included to the system take shape according to the choices, made via the user interface by the instructor. The course must be planned with an optimum content sequence for the students. For this reason, the system uses two different sizes of both content sets; first one is the smaller (called SDST) including approximately half of the content to produce a course of one semester, and the second one (called LDST) to produce a course of two semesters.

After the data for a course and its modules are provided properly, the system runs the genetic algorithm phase on these modules to optimize the sequence of the modules of the course. The number of chosen modules determines the length of the chromosomes used in the GA. For different training materials, the number of modules may differ; therefore, reason the chromosome sizes are not equal for each training program. Regardless from the length of the chromosome, the initial population of the GA is produced in the same way from the first module range obtained. In this study, there are two different sizes of the same dataset, having 17 and 30 modules among 39 modules of the whole training program. The same amount of modules for two different module range sizes (17 and 30) are also chosen from the control data, which has 42 modules. The prerequisite rules among the modules are the most important point making the sequencing process more difficult to manage. For this reason, two different sizes of the dataset are needed to analyze the performance of the GA.

While GA is performed, the fitness function of the GA to be applied to these chromosomes use a rule set that includes the prerequisite information of each module as mentioned above in contrast with the systems coding a rule as one chromosome consisting of several segments (Tseng, Chen, Hwang & Shen, 2008). The result of the GA produces the most reliable and useful range of the modules to be used in the compensation training.

An example to the module evaluation of the system for the training data set is given in Table 4.1. The chosen modules are selected among the total list of the modules and their duration is calculated. In the table, the numbers of chosen modules are shown. This data is important for the system because the initial population of the GA is generated from these module numbers.

This kind of a dataset has some handicaps. The modules to be chosen and the size of the chromosomes are not known at the beginning of the process. There are tight prerequisite rules among the modules. For this reason sequencing task has to take these rules into consideration. The tighter is the rules among the modules, the more difficult to obtain more reliable results.

Table 4.1 Modules Chosen for the Training Program

M#	8	5	23	2	17	4	6	7	12	20	29	3	9	10	18	28
Hours	4	2	2	6	4	4	16	8	4	6	2	12	4	6	4	6

CHAPTER FIVE
SYSTEM ANALYSIS AND DESIGN

Before performing the optimization process, the conditions and constraints of the problem should be stated clearly. In order to express the phases of the system explicitly, making a system analysis is a must. The system analysis includes the workflow of the system, design of the GA to be used with all its operators and planning the parameter tuning step. The workflow is an important phase because it can be considered as the route map to the project.

5.1 The Workflow of the System

The workflow for the project is explained in two major phases. In the first phase, a rule based genetic algorithm for module ranking is generated as given below in Figure 5.1. Fig 5.2 explains the details of the module sequencing process – the first step of Figure 5.1 - in more detail.

As the first phase, the rule based module sequencing software consists of gathering the needed data of modules as rules, finding the optimum range of modules by running the GA and producing raw data (runtime and fitness values, module ranges and data required for the further steps of the workflow) and XML outputs for the training program. The following phases are about the evaluation process of the results; that is, the obtained results are evaluated with a non – parametric correlation test and the best module range is determined. In detail, in the second phase, obtained results are routed to the rank correlation test phase. For evaluating the results, Spearman Rank Correlation Test is applied to the results and at the last step, the best module range is obtained. The parameter combination giving the best module range is also emphasized. The same process is also applied on the control data set to verify the results of the training data set.

The workflow of the system is applied in the same way to both the training data and the control data.

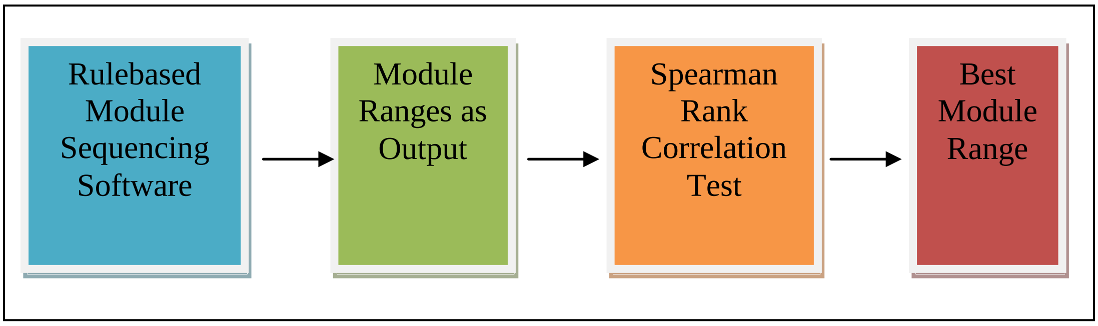

Rulebased Module Sequencing Software
Module Ranges as Output
Spearman Rank Correlation Test
Best Module Range

Figure 5.1 The general workflow of the system

The steps of the Rule Based Module Ranking Software is given below:

- The education material is divided into M modules.
- The time period and the prerequisite information of each module must be inserted into the system.
- Module names, module numbers and their durations reside in a database.
- The modules to be included to the curriculum plan are determined by the instructor via user interface according to the students' / trainees' demands or needs (Knowledge Acquisition phase in Figure 5.2.).
- The prerequisite rules of the modules are written in an XML file and attached to the software as XML.
- The first individual of the GA is formed by ranging the chosen module numbers randomly and the initial population is produced from the first individual.
- When the software is run, a module range is produced by the algorithm and a schedule for those modules is prepared and saved as an XML file.

Module sequencing process was being done by human experts manually before. Therefore, human experts naturally have a suggestion about how the correct sequence may be. The best module sequence obtained from the software should be compared with the suggestion of the human expert in order to be sure about its reliability. The number of the prerequisite rules among the modules is one of the factors that affect the reliability of the module range when compared with that of the human expert's.

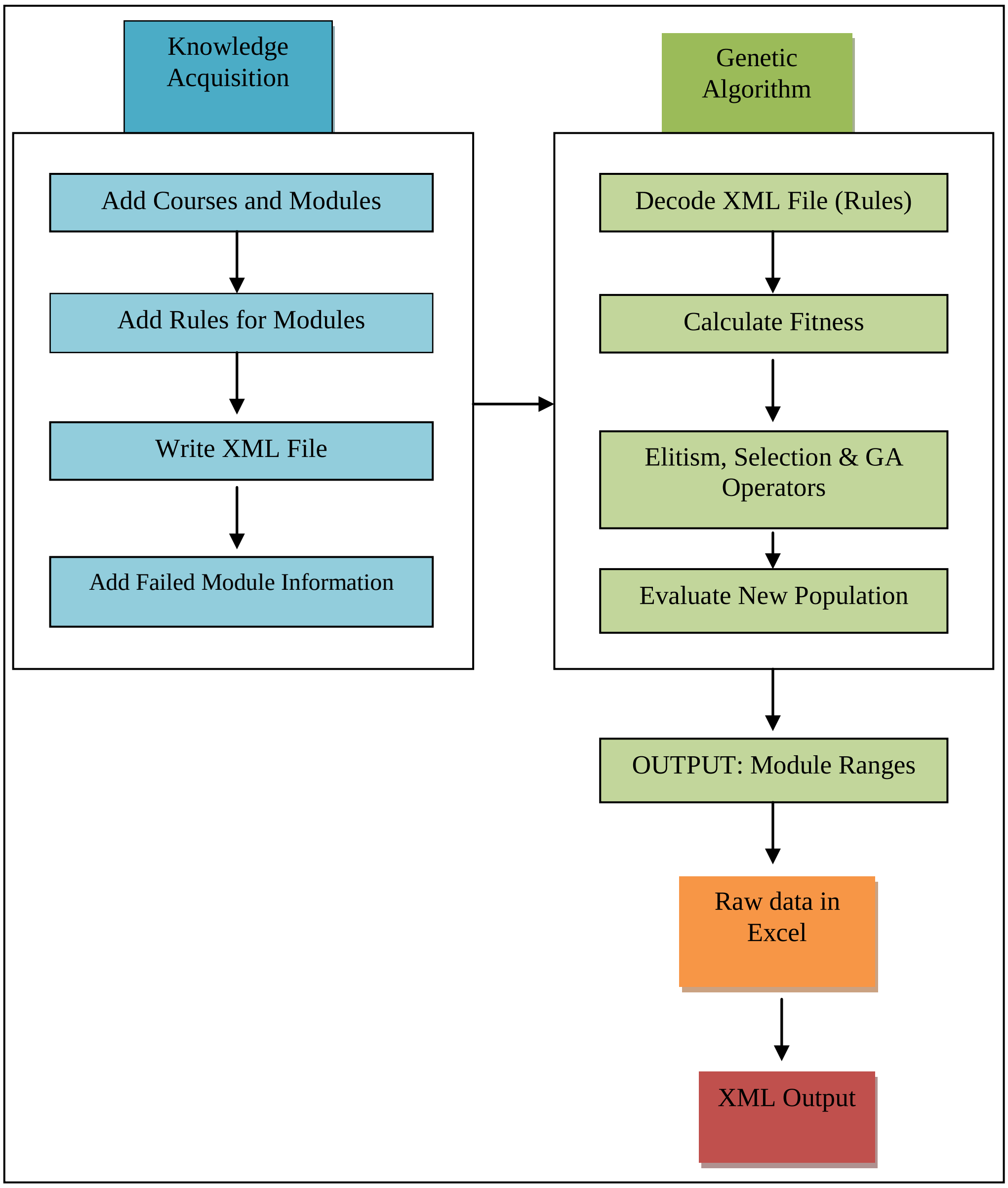


Figure 5.2 Detailed workflow for the Rule-based module ranking software

5.2 The Design of the Genetic Algorithm

The GA used in the project works on the chromosomes, which are composed of the module numbers. Each module number can take place in a chromosome only once; therefore *permutation encoding* technique is used for chromosome encoding (Julstrom, 2009). The module numbers that build up the chromosome are taken from database, but the prerequisite rules of the selected modules are taken from the XML file.

In the rule based GA stated in the thesis, which adopts the Pittsburg approach (Rodriguez, Escalante & Peregrin, 2011), there is only one rule behind each gene and the genes forming the chromosome represent a possible solution to the sequencing problem. For this reason, when a chromosome is formed from the selected modules of a course, a subset of the rule base is included to the system and GA only deals with the rules of the selected genes. Some modules may not be selected for the sequencing process; therefore the rules of the modules which are not selected are not included to the subset of the rule base.

A module may not have a prerequisite module. In that case, if the module does not have any prerequisites, then there is no rule about the module in the rule subset. The modules without prerequisites are easier to manage. Sequencing them is even easier, unless they are prerequisites for other modules.

5.2.1 Initial Population

The initial population of the GA is generated by using the module numbers to be ranged. These module numbers are ranged randomly (Derakhshi & Zandi, 2010) to form the first individual of the initial population. Then, by using the module numbers of the first individual, n different module ranges are generated for a population of n individuals. In the thesis, different sizes of populations with values of 100, 120, 140, 160, 180 and 200 are used.

5.2.2 GA Operators

The operators which are used in the project are explained briefly below. Three different types of crossover operators, two different types of mutation operators and a selection operator are in question in the thesis. Elitism is also applied.

5.2.2.1 Crossover

In the project, a standard GA is used with one – point order crossover, two – point crossover (Picek & Golub, 2010) and PMX. In one – point order crossover, the parents change their genetic material according to the randomly chosen splitting point. But since permutation encoding is used, the exchanging parts of the chromosomes are arranged considering the nonrecurring of the genes. In two-point crossover, the genes between two randomly chosen positions on the chromosome are inversed to obtain two new children for the next generation. PMX is an improved version of two – point order crossover, exchanging the genes between randomly chosen intervals and repairing the gene to prevent the repetitions.

The three crossover techniques are used in four different ways; we call them OX, OX2, PMX and OX' respectively.

- OX, 1-point order crossover of Davis (1991),
- OX2, 2 –point order crossover of Coley (1998),
- PMX, partially matched crossover (Sivanandam & Deepa, 2008),
- OX', one of the three crossover methods (OX, OX2, PMX) chosen randomly in each generation (Tseng, Chen, Hwang & Shen, 2008).

5.2.2.2 Mutation

Inversion mutation and swap mutation is applied in various probabilities as the mutation method in the study. The mutation is applied in low mutation probability values not to destroy the evolution period of the chromosomes. Using swap mutation and inversion mutation in various probabilities helps to manage the parameter tuning task of the GA.

When a chromosome is mutated with swap mutation, randomly chosen two genes (module numbers in this case) are swapped and a new child for the population is produced. With inversion mutation, an interval of genes is chosen and inversed. The aim of the mutation process is to sustain the diversity of the population and prevent the fitness values converge rapidly to maximum.

5.2.2.3 Selection

The selection operator is determined as the Linear Rank Selection operator (Greffenstette & Baker, 1989) as given in (1).

$$rankval_i = \frac{1}{N}\left(\min + \frac{(\max - \min)(rank_i - 1)}{N - 1} \right) \tag{1}$$

where *rankval_i* represents the rank value of the i^{th} individual in the population,
N represents the population size,
$rank_i$ represents the order of the i^{th} individual,
max + min = 2 and 1 ≤ max ≤ 2 (Chakraborty & Chakraborty, 1997).

5.2.2.4 Elitism

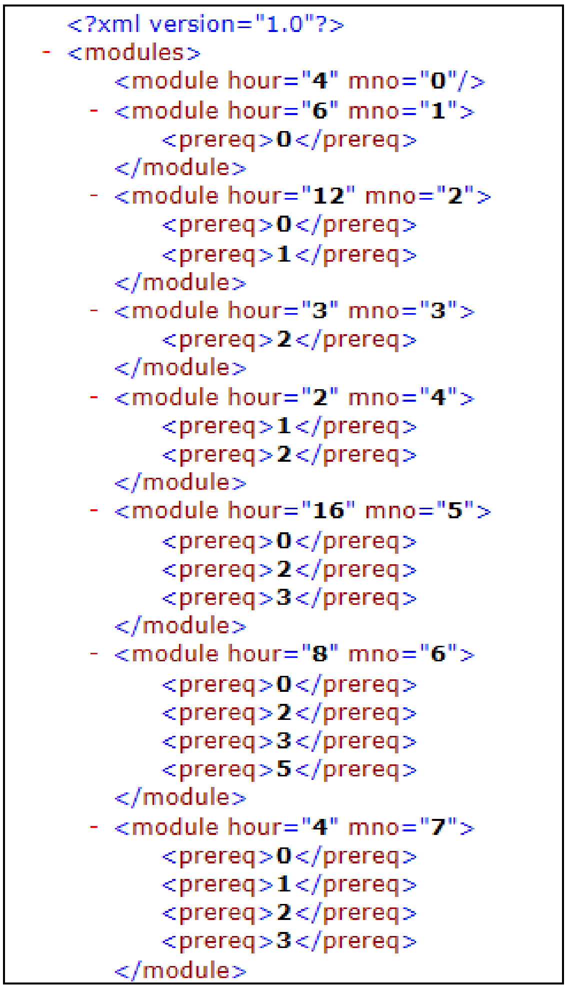
The elitism mechanism is also applied to protect the best individual in the population by passing a group of best individuals directly to the next generation (Cheng, Shi, Yin & Li, 2011). According to the elitist strategy, our system transfers 10% of the population having the best fitness values directly to the next generation. This ratio carries the individuals having the best fitness values to the next generation without being exposed to any genetic operators.

5.2.3 The Fitness Function

The input data of the fitness function of the project is the XML file in which the prerequisite rules of the modules are saved. The XML file including the rules is given in Figure 5.3. A more detailed XML files of the rules for the training data and the control data are given in Appendix C.

The logical representation of module rules resides in an XML file and then the

XML file is converted to a sparse matrix, where each row represents a module and the elements of a row consist of 0s. If the module has prerequisites, then the column representing the prerequisite module number is set to 1. Too many 1s in a row means that the module is more tightly related to its prerequisites.

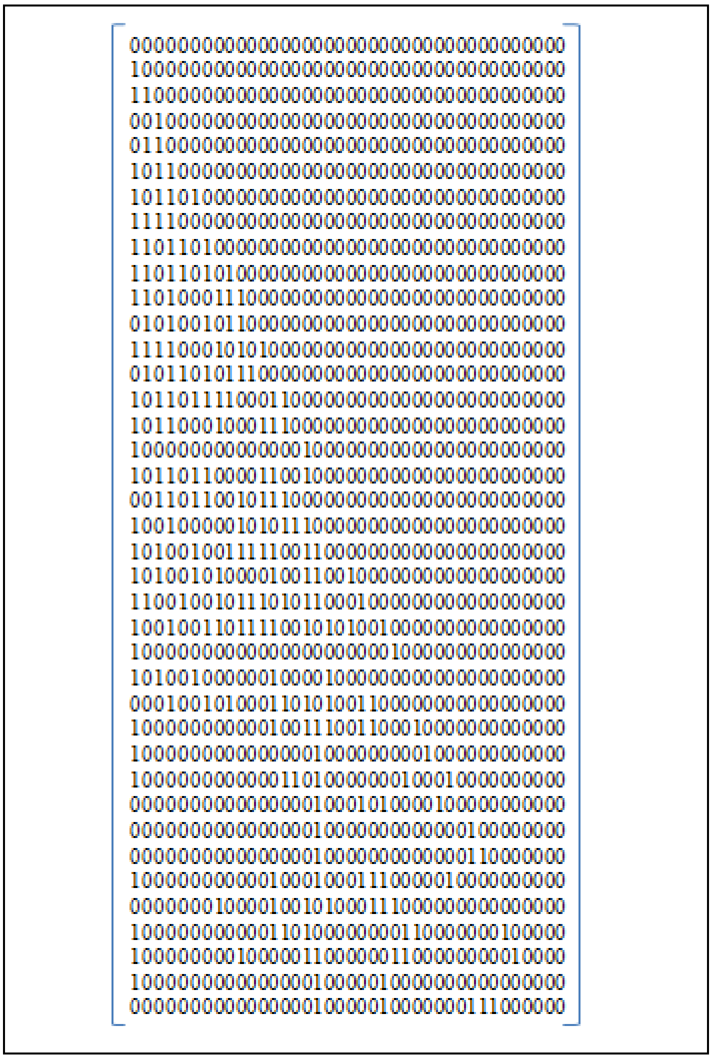


```
  <?xml version="1.0"?>
- <modules>
    <module hour="4" mno="0"/>
  - <module hour="6" mno="1">
      <prereq>0</prereq>
    </module>
  - <module hour="12" mno="2">
      <prereq>0</prereq>
      <prereq>1</prereq>
    </module>
  - <module hour="3" mno="3">
      <prereq>2</prereq>
    </module>
  - <module hour="2" mno="4">
      <prereq>1</prereq>
      <prereq>2</prereq>
    </module>
  - <module hour="16" mno="5">
      <prereq>0</prereq>
      <prereq>2</prereq>
      <prereq>3</prereq>
    </module>
  - <module hour="8" mno="6">
      <prereq>0</prereq>
      <prereq>2</prereq>
      <prereq>3</prereq>
      <prereq>5</prereq>
    </module>
  - <module hour="4" mno="7">
      <prereq>0</prereq>
      <prereq>1</prereq>
      <prereq>2</prereq>
      <prereq>3</prereq>
    </module>
```

Figure 5.3 XML file including rules among the training modules

The number of modules to be sequenced determines the length of the chromosomes used in the GA. The initial population of the GA is produced from the first module range obtained. While GA is performed, the fitness function of the GA to be applied to these chromosomes use a rule set that includes the prerequisite information of each module as mentioned above.

The fitness function mechanism works by parsing the 1s in the prerequisite matrix and the penalty scores calculated for the modules on the chromosome. It is quite practical way of transferring the rules to a mathematical representation. Saving the rules in a matrix has a few benefits. The system does not have to execute any queries from database to acquire the rule data. It is easier to save and process data when it is in a sparse matrix format. An XML file including the rules can be converted to a sparse matrix easily. An example of the sparse matrix is given in Figure 5.4.



```
000000000000000000000000000000000000000
100000000000000000000000000000000000000
110000000000000000000000000000000000000
001000000000000000000000000000000000000
011000000000000000000000000000000000000
101100000000000000000000000000000000000
101101000000000000000000000000000000000
111100000000000000000000000000000000000
110110100000000000000000000000000000000
110110101000000000000000000000000000000
110100011100000000000000000000000000000
010100101100000000000000000000000000000
111100010101000000000000000000000000000
010110101110000000000000000000000000000
101101111000110000000000000000000000000
101100010001110000000000000000000000000
100000000000000100000000000000000000000
101101100001100100000000000000000000000
001101100101110000000000000000000000000
100100000101011100000000000000000000000
101001001111100110000000000000000000000
101001010000100110010000000000000000000
110010010111010110001000000000000000000
100100110111100101010010000000000000000
100000000000000000000001000000000000000
101001000000100001000000000000000000000
000100101000110101001100000000000000000
100000000000100111001100010000000000000
100000000000000010000000001000000000000
100000000000011010000000100010000000000
000000000000000010001010000100000000000
000000000000000010000000000000100000000
000000000000000010000000000000110000000
100000000000100010001110000010000000000
000000010000100101000111000000000000000
100000000000110100000000110000000100000
100000000100000110000001100000000010000
100000000000000010000010000000000000000
000000000000000010000010000000111000000
```

Figure 5.4 Sparse matrix representation of the XML files including the rules

When a prerequisite of a module appears after that module in the chromosome, the penalty point is increased by calculating with a weight value of the module. Weight values for each module are calculated according to the number of prerequisites. The module with more prerequisites is more difficult to locate in the curriculum plan. The pseudo code written for the fitness function is given in Figure 5.5.

When the penalty score calculation for the ith chromosome ended, the fitness value of the chromosome is calculated by using (2).

$$fitness[i] = \frac{1}{\sum_{k=1}^{m} \frac{penaltycount_k}{prereqcount_k} + 1} \tag{2}$$

where *m* is the number of modules in a chromosome, *penaltycount* is the number of prerequisite modules for module *k* exisiting in the chromosome and *prereqcount* is the total number of prerequisite modules for module *k*.

```
for each individual i do
begin
   penaltypoint = 0;
   for each module in the chromosome do
   begin
      penaltycount = 0;
      for each module after itself do
      begin
         if a prerequisite module exists after itself
         then penaltycount$_i$++;
      end
      penaltyratio$_i$= penaltycount$_i$/total prereqcount$_i$
      penaltypoint = penaltypoint + penaltyratio$_i$
   end
   fitnessvalue$_i$ = (1/ (penaltypoint+1))
end
```

Figure 5.5 Pseudo code for calculating the fitness function

While calculating the penalty point for each individual according to the prerequisites of the modules on the chromosome, the lower penalty point shows that the fitness value of the individual is higher. Having a higher fitness value means the individual is more suitable as the solution of the problem. On the other hand, the chromosome with the highest penalty score is defined as the worst individual of the population according to Formula (2). GA tries to minimize the penalty scores and maximize the fitness values (Hung, 2009). In the fitness functions in which penalty scores are calculated, the aim is always minimizing the penalties and maximizing the fitness value (Zheng, Liu, Geng & Yang, 2009). The chromosome with the minimum value of the penalty score turns out to be the best individual of the generation. As required as a rule of fitness evaluation, the individuals are then sorted from the worst fitness value to the best.

5.3 Parameter Tuning

In order to obtain the best results from the GA, parameter tuning is done by changing the three basic parameters of the GA. These parameters are:

- Number of generation (maxgen),
- Crossover rate (*crate*) and
- Mutation rate (*mrate*).

The parameters used for different scenarios (S1 – S63) are given in Table 5.2. These parameters are applied to all four GAs (OX, OX2, PMX and OX') for six different population sizes. The same is applied on both a smaller dataset having 17 genes in a chromosome and a larger dataset having 30 genes in a chromosome.

The curriculum sequencing is done for the cases, in which all the modules of a whole training / course are not included to the system. The fitness function of the genetic algorithm has been improved and tested with new parameter combinations given in Table 5.1.

Table 5.1 GA Parameters for Different Scenarios

Parameter	Value
Population Size	100, 120, 140, 160, 180, 200
# of Generations	500, 750, 1000
Crossover Rate (crate)	0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1
Mutation Rate (mrate)	0.1, 0.15, 0.2

Briefly, the GA of the software is tested for:

- 2 datasets of different size (which are called *SDST* and *LDST* for the smaller and larger ones respectively),
- 6 different population sizes (given in Table 5.1),
- 4 crossover operators
- 63 scenarios for each operator (3 generation values, 7 crossover rate values and 3 mutation rate values)
- 20 run for each scenario.

Although mutation is naturally included into the parameter tuning process with 3 different mutation rate values, the two different mutation operators are used randomly in all of the executions. This means that, swap mutation and inversion mutation have the same amount of effects on the scenarios.

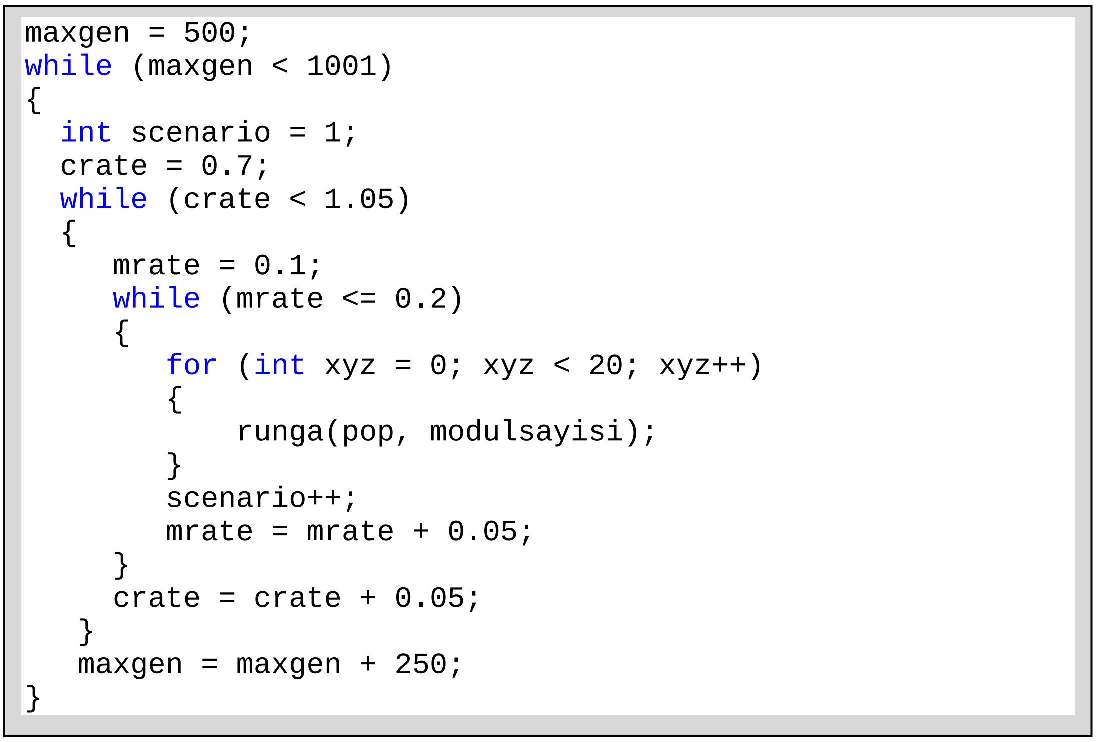
Table 5.2 Parameter Tuning with the scenarios for 500 (S1 – S21), 750 (S22 – S42) and 1000 (S43 – S63) generations respectively.

Scenario	crate	mrate	Scenario	Crate	mrate	Scenario	Crate	Mrate
S1	0.70	0.1	**S22**	0.70	0.1	**S43**	0.70	0.1
S2	0.70	0.15	**S23**	0.70	0.15	**S44**	0.70	0.15
S3	0.70	0.2	**S24**	0.70	0.2	**S45**	0.70	0.2
S4	0.75	0.1	**S25**	0.75	0.1	**S46**	0.75	0.1
S5	0.75	0.15	**S26**	0.75	0.15	**S47**	0.75	0.15
S6	0.75	0.2	**S27**	0.75	0.2	**S48**	0.75	0.2
S7	0.80	0.1	**S28**	0.80	0.1	**S49**	0.80	0.1
S8	0.80	0.15	**S29**	0.80	0.15	**S50**	0.80	0.15
S9	0.80	0.2	**S30**	0.80	0.2	**S51**	0.80	0.2
S10	0.85	0.1	**S31**	0.85	0.1	**S52**	0.85	0.1
S11	0.85	0.15	**S32**	0.85	0.15	**S53**	0.85	0.15
S12	0.85	0.2	**S33**	0.85	0.2	**S54**	0.85	0.2
S13	0.90	0.1	**S34**	0.90	0.1	**S55**	0.90	0.1
S14	0.90	0.15	**S35**	0.90	0.15	**S56**	0.90	0.15
S15	0.90	0.2	**S36**	0.90	0.2	**S57**	0.90	0.2
S16	0.95	0.1	**S37**	0.95	0.1	**S58**	0.95	0.1
S17	0.95	0.15	**S38**	0.95	0.15	**S59**	0.95	0.15
S18	0.95	0.2	**S39**	0.95	0.2	**S60**	0.95	0.2
S19	1.00	0.1	**S40**	1.00	0.1	**S61**	1.00	0.1
S20	1.00	0.15	**S41**	1.00	0.15	**S62**	1.00	0.15
S21	1.00	0.2	**S42**	1.00	0.2	**S63**	1.00	0.2

Here GA is run for 63 different parameter combinations. These combinations are shown in Table 5.2 (S1 – S63). When GA is executed for these different parameter combinations and crossover techniques, each combination for 20 times (Roeva, 2008), 24 different results sets are obtained for both SDST and LDST. Each result set includes 63 different solutions, since there are 63 scenarios tested. All of these solutions are included to the evaluation process without eliminating any of them manually. The parameter combination, which gives the most reliable module range, is accepted as the most appropriate combination to be used in sequencing problems about curriculum planning.

Executing the same parameter combination 20 times guarantees to find the best solution. Also calculating the average values for these runs of the same scenario gives an idea about the success of the paramaters. Each set of results are compared with the human-expert's suggestion by testing them with a non-parametric correlation test. The result ranges are tested with Spearman Rank Correlation to decide their reliability. Among these, best module ranges and the number of reliable ranges are obtained. The parameter combination having the most reliable results is also important from the parameter tuning point of view where the number of reliable module ranges is also important as how reliable they are.

These combinations are very time consuming to handle one by one. The results are a great amount to manage when all 1260 runs for each GA is done manually. Instead, parameter tuning process is automatized within the software. In previous studies, automated parameter tuning was done for only one parameter at a time (Liu, Mernik & Bryant, 2004). In this study, all values of crossover and mutation operators and number of generation values are tuned together and it takes a long time for the computer find 1260 different results. For this reason, except the module range results, the system also calculates the execution time for each automated process. The algorithm for automated parameter tuning done in the study is given in Figure 5.6.



```
maxgen = 500;
while (maxgen < 1001)
{
  int scenario = 1;
  crate = 0.7;
  while (crate < 1.05)
  {
     mrate = 0.1;
     while (mrate <= 0.2)
     {
        for (int xyz = 0; xyz < 20; xyz++)
        {
            runga(pop, modulsayisi);
        }
        scenario++;
        mrate = mrate + 0.05;
     }
     crate = crate + 0.05;
   }
   maxgen = maxgen + 250;
}
```

Figure 5.6 Algorithm for automated parameter tuning of GA is given. It performs 1260 runs at the same time with 3 different generation values, 7 different crossover values and 3 different mutation values.

5.4 Spearman Rank Correlation

When a solution to a sequencing problem is found, it is important to show the reliability of the results to verify that the study has a scientific value. The statistical analysis is used in such cases to convince people about the trustfullness of the results. When the GA is run with the given features and parameters, n module ranges for n individuals of the population are obtained as follows:

$\text{Indv}_1 \rightarrow$ 1-32-4-8-9-14-10-13-6-26-15-16-20-23-35-29-38

$\text{Indv}_2 \rightarrow$ 4-1-9-35-38-6-10-14-8-13-15-16-32-23-29-26-20

…

$\text{Indv}_n \rightarrow$ 4-1-38-26-9-10-14-8-13-29-16-23-6-32-15-20-35

The reliability of these module ranges must be tested by comparing with a range given by an expert (The instructor of the training in this case). It is possible to make any judgements about the ranges only after applying these tests. The test to be applied for this purpose is Spearman Rank Correlation, which is a non – parametric test used in statistical analysis. It is used in cases where testing the reliability of the range of data is more important than the numerical values of the data (Sheskin, 2000). Formula (3) is used to apply the Spearman Rank Correlation test.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \tag{3}$$

where ρ indicates Spearman Rank Correlation coefficient, d_i indicates the difference between the expected and observed rank values and n is the number of alternatives (observations). The hypothesis is set as given below:

H_0: There is no correlation between the two ranges.

H_1: There is a correlation between the two ranges.

Using ρ values, t values are calculated with the Formula (4).

$$t = \frac{\rho}{\sqrt{(1-\rho^2)/(n-2)}} \tag{4}$$

For each parameter combination, 20 t values are calculated because 20 different module ranges are obtained with each single parameter combination by executing the software 20 times.

After calculating the t values, they are compared with the critical value of the t Table (t – Table used in this study is given in Appendix E (Bissonette, 2011)). If the t value of a range is greater than the critical value given in the table, then hypothesis H_0 can be rejected and it can be said that the range is reliable. While determining the correct t value to be compared from t Table, the Degree of Freedom (DF) and Confidence Level must be clearly decided.

DF = n – 2, where n, in this case, is the number of modules included in the range. Confidence Level is the level of probability value, in which the results can be accepted as “reliable”. In the study, the confidence level is set to 99%, therefore, the t value to be observed in t – Table must be found according to $\alpha = 0.01$.

The result set containing the module ranges is written in an MS Excel file and the file is parsed and read by the software implemented (called Curriculum Organizer) to make the evaluation process. With the software, the ρ and t values are calculated automatically for all solution sets. It is implemented in MS Visual Studio 2008 environment with C# and is able to work on MS Excel 2007 files. The calculated ρ and t values are saved in another MS Excel file.

They assess the reliability of the ranges; the t values are tested according to the criteria given in Table 5.3.

Table 5.3 Spearman Evaluation Criteria

Number of Modules (n)	17	30
Degree of Freedom (n-2)	15	28
Tolerance	1%	
Confidence Level (p) (two tailed)	0.005	0.005
T value	2.947	2.763

The reliability of the module ranges are evaluated with a tolerance percentage of 1% ($p<0.01$). According to the hypothesis, to be able to understand whether there is a correlation between the output and the expert suggestion, the t values must be compared with the value of 2.947 for SDST and with the value of 2.763 for LDST.

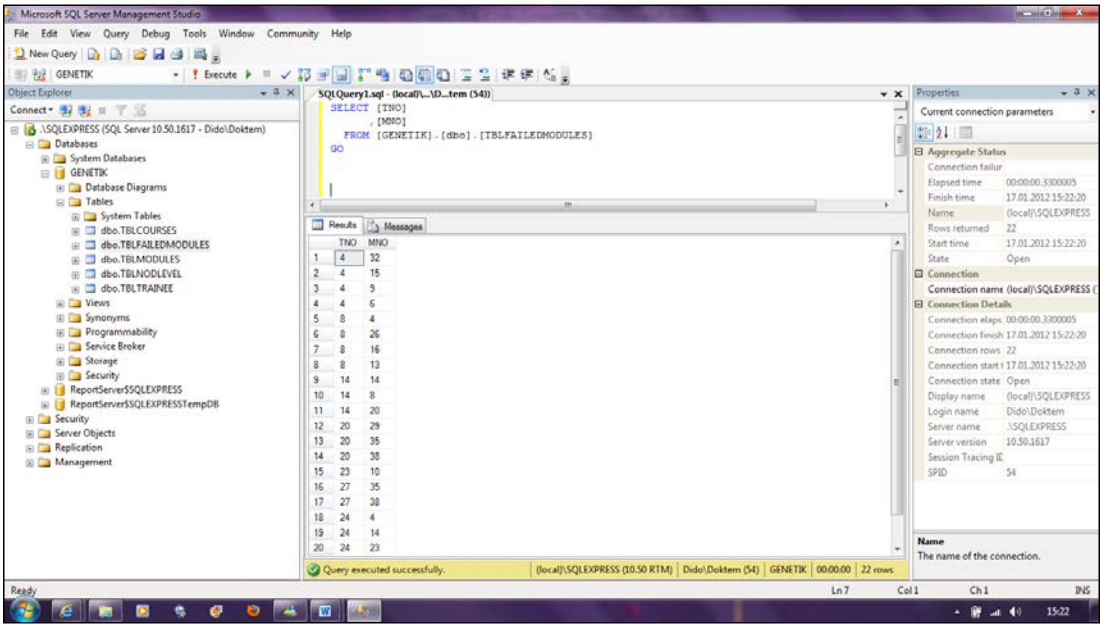
CHAPTER SIX
PROJECT IMPLEMENTATION

6.1 The Software Environment

The project has been developed in Microsoft Visual Studio 2008 platform with Microsoft Visual C# 2008 programming language. Microsoft SQL Server 2008 is used to manage the database tasks of the system. The experimental results are gathered with an Intel Core i7 2630QM 2.00 GHz/8GB computer.

6.2 The Database Design

The database design of the project is constructed with Microsoft SQL Server 2008 R2 Management Studio, which the user interface is shown in Figure 6.1. The tables and the relations among the tables are explained below in detail.



```
SELECT [TNO]
      ,[MNO]
  FROM [GENETIK].[dbo].[TBLFAILEDMODULES]
GO
```

	TNO	MNO
1	4	32
2	4	15
3	4	9
4	4	6
5	8	4
6	8	26
7	8	16
8	8	13
9	14	14
10	14	8
11	14	20
12	20	29
13	20	35
14	20	38
15	23	10
16	27	35
17	27	38
18	24	4
19	24	14
20	24	23

Query executed successfully. | (local)\SQLEXPRESS (10.50 RTM) | Dido\Doktem (54) | GENETIK | 00:00:00 | 22 rows

Aggregate Status	
Connection failur	
Elapsed time	00:00:00.3300005
Finish time	17.01.2012 15:22:20
Name	(local)\SQLEXPRESS
Rows returned	22
Start time	17.01.2012 15:22:20
State	Open
Connection	
Connection name	(local)\SQLEXPRESS (
Connection Details	
Connection elaps	00:00:00.3300005
Connection finish	17.01.2012 15:22:20
Connection rows	22
Connection start t	17.01.2012 15:22:20
Connection state	Open
Display name	(local)\SQLEXPRESS
Login name	Dido\Doktem
Server name	.\SQLEXPRESS
Server version	10.50.1617
Session Tracing IE	
SPID	54

Name
The name of the connection.

Figure 6.1. SQL Server 2008 R2 Management Studio environment

6.2.1 The Database Tables

The data about the courses to be optimized is kept in table TBLCOURSES. The primary key of this table (CNO) is used in other tables to distinguish the modules of different courses.

Table 6.1 TBLCOURSES

Column	**Data Type**	**Constraints**
CNO	Smallint	PK
Name	Varchar(50)	NOT NULL
Description	Varchar(100)	

The data about the modules of the courses reside in table TBLMODULES. The primary key of this table (MNO) represents the modules in a chromosome. For this reason, the data in this table has great importance for the software. The data in “hour” column is used to calculate the duration of the sequenced modules.

Table 6.2 TBLMODULES

Column	**Data Type**	**Constraints**
MNO	Smallint	PK
NAME	Varchar(100)	NOT NULL
Prerequisite	Smallint	
Hour	Smallint	
Lıd	Smallint	
CNO	Smallint	Foreign key, References (CNO) on TBLCOURSES

The data about the trainees of the courses reside in table TBLTRAINEE. The primary key of this table (TNO) is used to determine the failed modules of the trainees.

Table 6.3 TBLTRAINEE

Column	Data Type	Constraints
TNO	Smallint	PK
Name	Varchar(50)	NOT NULL
TotalNet	Smallint	
Lıd	Smallint	

The data about the trainee – failed module pairs reside in table TBLFAILEDMODULES. The foreign keys of this table (TNO and MNO) are used to determine the failed modules of the trainees. When the module numbers are distinctly selected from the table, the modules to be ranged with the GA is obtained. For this reason, the data in this table has a great importance for the system.

Table 6.4 – TBLFAILEDMODULES

Column	Data Type	Constraints
TNO	Smallint	Foreign key, References (TNO) on TBLTRAINEE
MNO	Smallint	Foreign key, References (MNO) on TBLMODULES

The scheme showing the relations among the tables of the database is given in Figure 6.2.

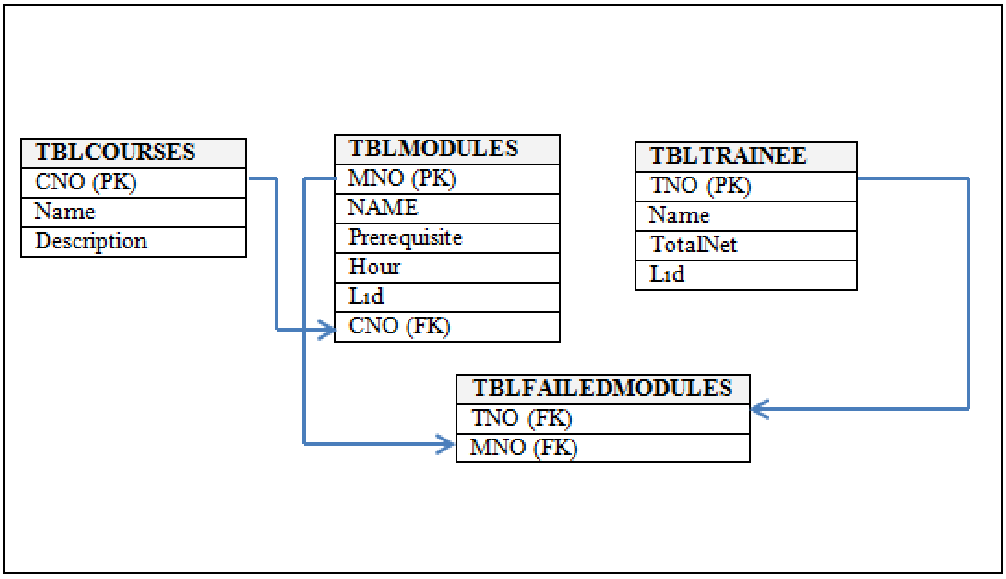


Figure 6.2. Relations among the tables

6.3 The Module Ranking Software

The project has a user-friendly UI, where the instructors can insert rules about the modules of a course or add/drop courses and modules (Fig 6.3).

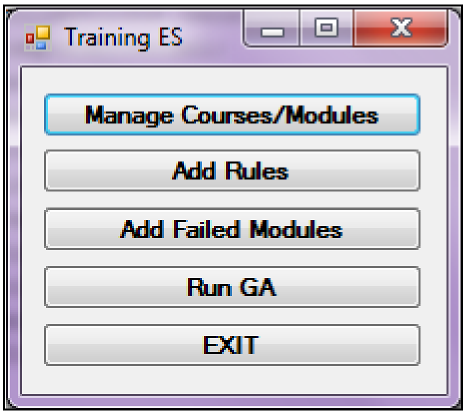
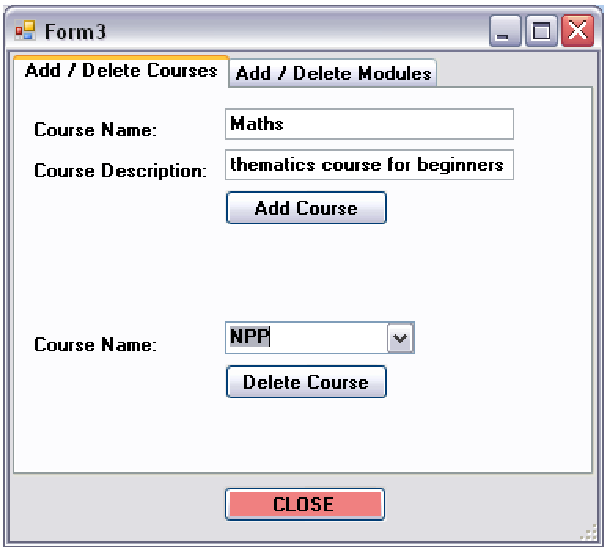


Figure 6.3. Main menu

There are three steps to be handled in the program before the genetic algorithm is run. The users of this project are expected to be the instructors of a course to be given. So, they have to add or delete the courses to the system with their modules. The users can add or delete courses and add or delete modules to the chosen course from the second form opened (Figure 6.4 and Figure 6.5).

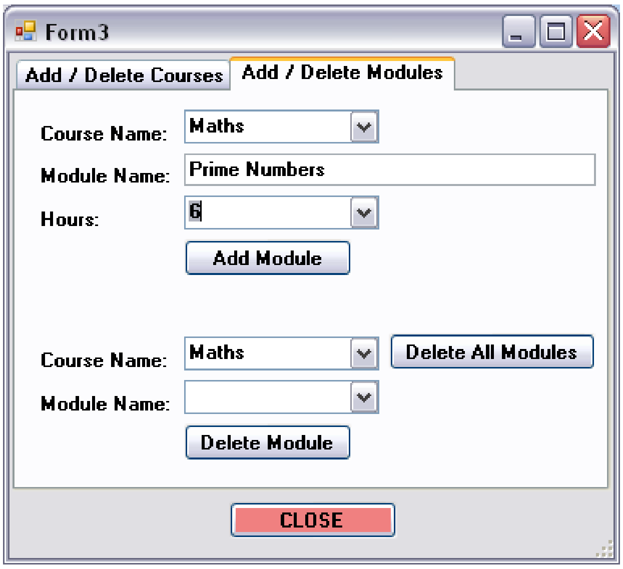


Course Name: Maths

Course Description: thematics course for beginners

Course Name: NPP

Figure 6.4 Add / Delete Courses



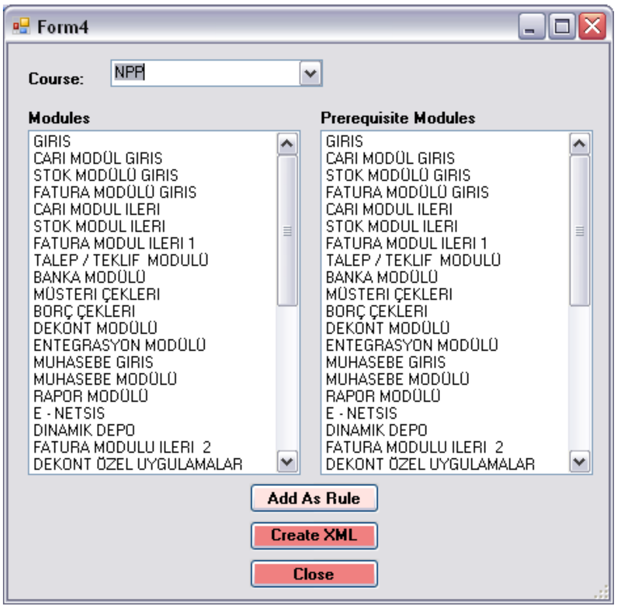
Course Name: Maths
Module Name: Prime Numbers
Hours: 6

Course Name: Maths
Module Name:

Figure 6.5 Add / Delete Modules

The database of the project is directly affected by the added or deleted courses or modules. When the user closes this form, he is guided to the main form again. With the second button on the main menu, the user can define the rules of a course by determining the prerequisite modules of each module included to that course. To add a rule, the user selects a module from the first list on the left and can choose more than one modules from the list on the right at the same time as the prerequisite of the module. When "Add as Rule" button is pressed (Figure 6.6), the chosen module and prerequisite information is saved to a matrix. After defining all prerequisites for all modules on the list, the user presses "Create XML" button. At this step, the program forms an XML file by using the prerequisites information gathered in the matrix.

After defining the rules, the user can choose the unsuccessful modules of each user, which the new training program will be composed of. When a trainee is chosen, the modules that the trainee is responsible for appear. The user chooses the modules in which the trainee was unsuccessful and presses the "Add to Working Memory" button. This is another aspect of knowledge acquisition for the system (Figure 6.7). If the user needs, he can clear the previous data given to the system with the "Clear Previous Data" button.



Course: NPP

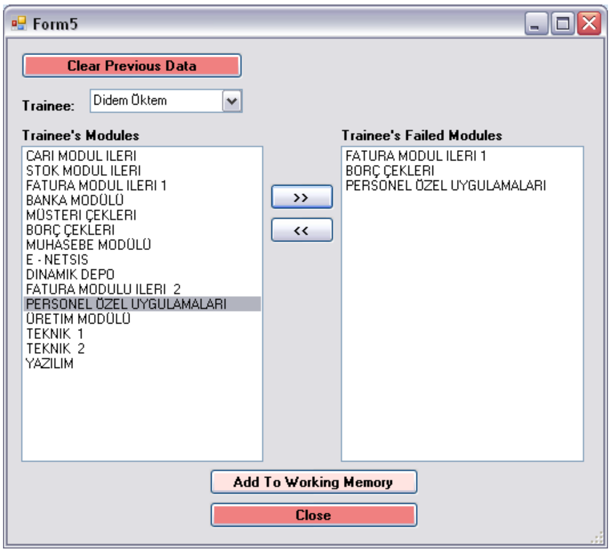
Modules

GIRIS
CARI MODÜL GIRIS
STOK MODÜLÜ GIRIS
FATURA MODÜLÜ GIRIS
CARI MODUL ILERI
STOK MODUL ILERI
FATURA MODUL ILERI 1
TALEP / TEKLIF MODULÜ
BANKA MODÜLÜ
MÜSTERI ÇEKLERI
BORÇ ÇEKLERI
DEKONT MODÜLÜ
ENTEGRASYON MODÜLÜ
MUHASEBE GIRIS
MUHASEBE MODÜLÜ
RAPOR MODÜLÜ
E - NETSIS
DINAMIK DEPO
FATURA MODULU ILERI 2
DEKONT ÖZEL UYGULAMALAR

Prerequisite Modules

GIRIS
CARI MODÜL GIRIS
STOK MODÜLÜ GIRIS
FATURA MODÜLÜ GIRIS
CARI MODUL ILERI
STOK MODUL ILERI
FATURA MODUL ILERI 1
TALEP / TEKLIF MODULÜ
BANKA MODÜLÜ
MÜSTERI ÇEKLERI
BORÇ ÇEKLERI
DEKONT MODÜLÜ
ENTEGRASYON MODÜLÜ
MUHASEBE GIRIS
MUHASEBE MODÜLÜ
RAPOR MODÜLÜ
E - NETSIS
DINAMIK DEPO
FATURA MODULU ILERI 2
DEKONT ÖZEL UYGULAMALAR

Figure 6.6 Add Rules Screen for the Training Data



Trainee: Didem Öktem

Trainee's Modules

CARI MODUL ILERI
STOK MODUL ILERI
FATURA MODUL ILERI 1
BANKA MODÜLÜ
MÜSTERI ÇEKLERI
BORÇ ÇEKLERI
MUHASEBE MODÜLÜ
E - NETSIS
DINAMIK DEPO
FATURA MODULU ILERI 2
PERSONEL ÖZEL UYGULAMALARI
ÜRETIM MODÜLÜ
TEKNIK 1
TEKNIK 2
YAZILIM

Trainee's Failed Modules

FATURA MODUL ILERI 1
BORÇ ÇEKLERI
PERSONEL ÖZEL UYGULAMALARI

Figure 6.7 Determining the Chosen Modules of Users for the Training Data

After adding the unsuccessful modules for the trainees to the system, the user closes the form and turns back to the main menu. From the menu, the user presses the “Run GA” button and a new form is opened to run and view the results of the genetic algorithm as shown in Figure 6.8.

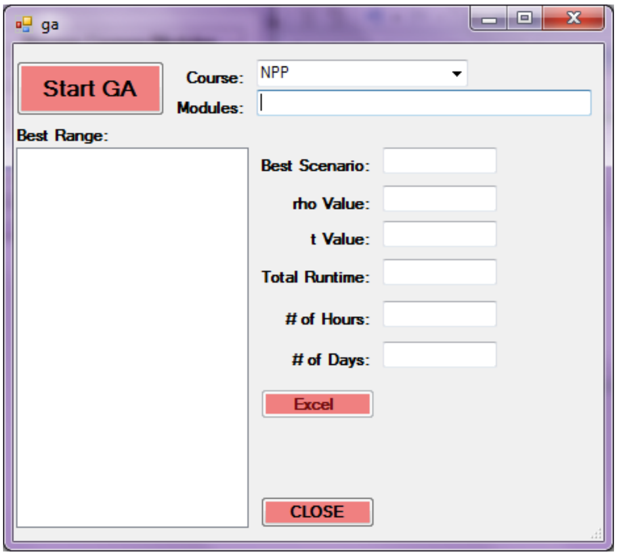
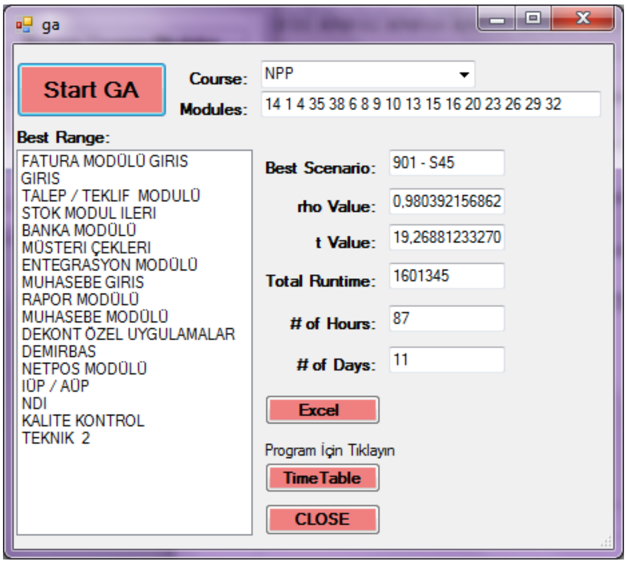


Figure 6.8 Running the GA – When “Start GA” button is clicked, the program executes 1260 times and writes all results to *.csv files. On the screen, only the best module range and its information is shown.

When the button “Start GA” is pressed, the program gives the numbers of the modules are listed from the most requested to the least. The total time period for the chosen modules is calculated. The resulting order of the modules are listed both above in the “Best Individual” text box with numbers only and in the list box to the right of the screen with their names. Fitness and population information given below are only to check whether the program has operated properly or not (Figure 6.9.). since the module ranges are saved as “*.csv” files, the module ranges as output are saved in “moduleranges.csv” file. An example of module ranges as a .csv file is given in Figure 6.10.

According to the total duration of the modules, an approximate duration for the schedule is calculated and shown in the “Estimated Days” text box. Runtime period is calculated in each run of the genetic algorithm and this runtime information is used to make an evaluation about the performance of different crossover methods used in the project. Since 1260 runs are handled at once, the runtime values are given in minutes.



Course: NPP

Modules: 14 1 4 35 38 6 8 9 10 13 15 16 20 23 26 29 32

Best Range:
FATURA MODÜLÜ GIRIS
GIRIS
TALEP / TEKLIF MODULÜ
STOK MODUL ILERI
BANKA MODÜLÜ
MÜSTERI ÇEKLERI
ENTEGRASYON MODÜLÜ
MUHASEBE GIRIS
RAPOR MODÜLÜ
MUHASEBE MODÜLÜ
DEKONT ÖZEL UYGULAMALAR
DEMIRBAS
NETPOS MODÜLÜ
IÜP / AÜP
NDI
KALITE KONTROL
TEKNIK 2

Best Scenario: 901 - S45

rho Value: 0,980392156862

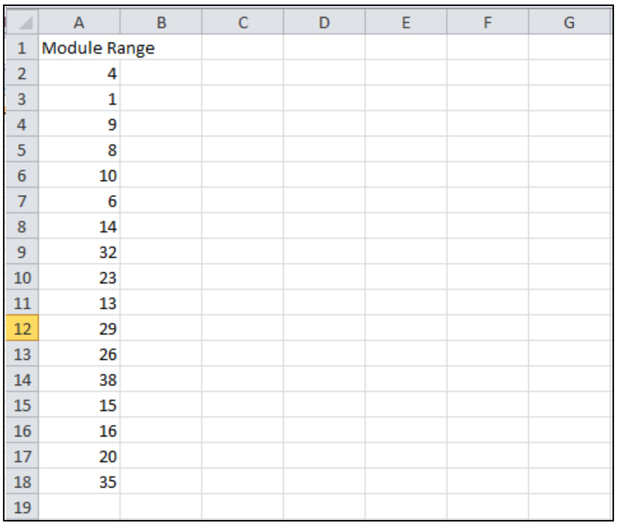
t Value: 19,26881233270

Total Runtime: 1601345

# of Hours: 87

# of Days: 11

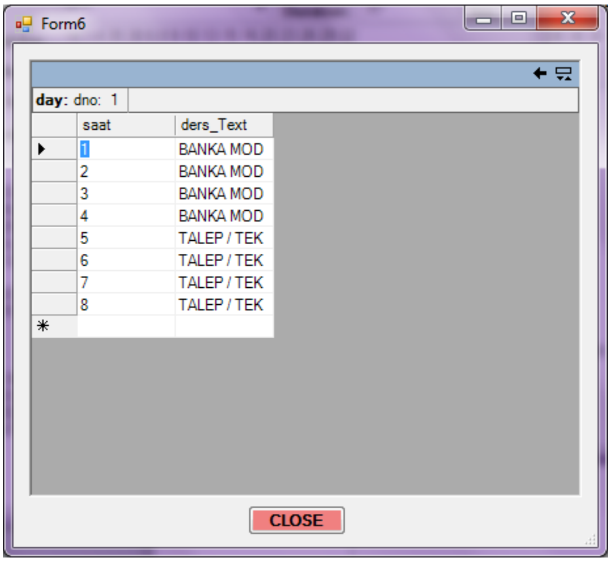
Figure 6.9 GA Results for the Training Data



	A	B	C	D	E	F	G
1	Module Range						
2	4						
3	1						
4	9						
5	8						
6	10						
7	6						
8	14						
9	32						
10	23						
11	13						
12	29						
13	26						
14	38						
15	15						
16	16						
17	20						
18	35						
19							

Figure 6.10 A sample module range in a .csv file (The numbers represent the modules)

The output course schedule is both written to an XML file and to a DataGrid object in the program. The DataGrid view for one of the obtained results is given in Figure 6.11. In the figure, the schedule is shown day by day, where it is not possible to show the whole schedule with DataGrid. For this reason, XML files seem to be a better solution to keep the output data. The XML file including the final timetable is given in Appendix D.



day: dno: 1

	saat	ders_Text
▶	1	BANKA MOD
	2	BANKA MOD
	3	BANKA MOD
	4	BANKA MOD
	5	TALEP / TEK
	6	TALEP / TEK
	7	TALEP / TEK
	8	TALEP / TEK
*		

Figure 6.11 The training program shown day by day

CHAPTER SEVEN
RESULTS

7.1 Runtime Values

Execution times for both the training and the control data for SDST and LDST are also given in Table 7.1(a), Table 7.1(b), Table 7.2(a) and Table 7.2(b) respectively.

Table 7.1(a) Execution times for SDST for the training data (min)

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	34.50	**19.41**	22.67	31.14
120	40.09	**23.94**	26.35	36.15
140	47.41	**28.76**	32.98	44.90
160	58.53	**33.35**	37.66	51.87
180	65.12	**39.24**	43.39	60.27
200	75.08	**43.53**	50.01	69.05

Table 7.1(b) Execution times for SDST for the control data (min)

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	32.88	**19.12**	22.19	23.23
120	40.02	**24.51**	26.47	28.63
140	47.68	**28.39**	32.97	35.42
160	56.75	**33.43**	37.40	41.08
180	66.15	**41.83**	43.59	46.96
200	71.45	**43.20**	48.14	52.28

For SDST, the execution time for OX2 is by far the best among other GAs for all population sizes for both of the training and control data sets. This is because of the performance of 2 – point order crossover is much better than the performances of 1 – point crossover and PMX. PMX is the second crossover operator performing well and when the three are used together, the execution times give the third best results in OX’.

Table 7.2(a) Execution times for LDST for the training data (min)

	GAs			
Pop. Size	OX	OX2	PMX	OX'
100	105.02	48.11	**42.63**	78.78
120	130.24	**51.89**	52.63	97.94
140	154.96	**60.63**	64.81	115.45
160	173.88	72.27	**71.93**	135.59
180	199.66	**79.74**	81.69	152.96
200	223.18	91.36	**94.37**	172.64

Table 7.2(b) Execution times for LDST for the control data (min)

	GAs			
Pop. Size	OX	OX2	PMX	OX'
100	107.38	**42.94**	44.05	56.27
120	125.85	**50.32**	51.67	70.73
140	148.13	61.77	**60.97**	83.00
160	173.87	**68.66**	70.68	88.48
180	204.03	80.64	**80.05**	104.93
200	215.08	91.11	**89.95**	120.36

For LDST, the execution times for OX2 and PMX are better than OX and OX'. There is not an obvious difference between OX2 and PMX for a larger dataset. Both the training and the control data sets give similar results, in which the results of the control data set verify the results of the training data set.

Comparative runtime graphs of the training data, which show the effect of data set growth for 100 individuals for both the training and the control data are given in Figure 7.1(a) and Figure 7.1(b) respectively. The runtime graphs of other population sizes are listed in Appendix P.

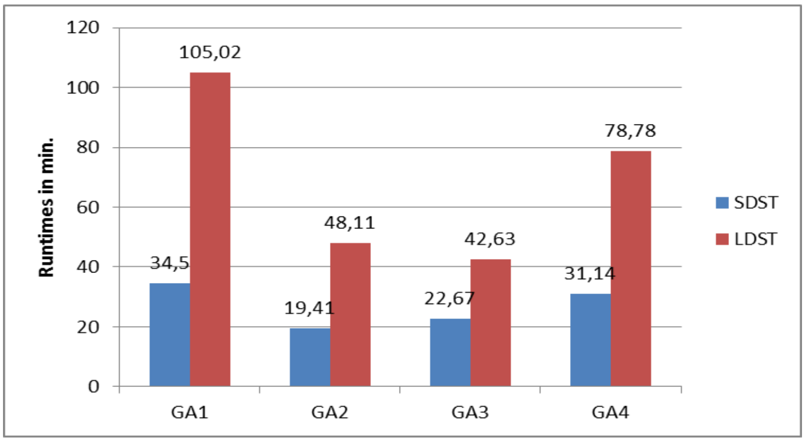


Figure 7.1(a) Runtime graphs showing the dataset growth of training data for 100 individuals.

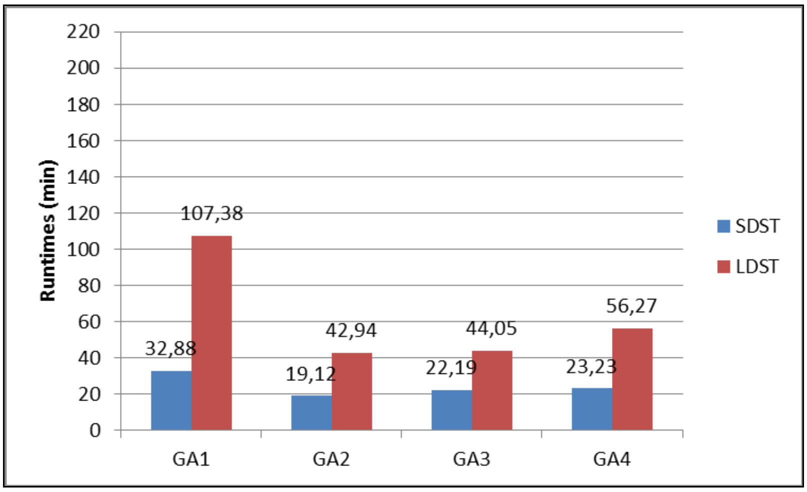


Figure 7.1(b) Runtime graphs showing the dataset growth of control data for 100 individuals.

Execution times for the larger dataset show that, increasing the dataset size almost by two does not affect the performance of PMX much. For OX and OX2, although OX2 performs the best for SDST, the execution times almost tripled when compared to the results of the smaller dataset. But execution times for PMX increased only for two times for the smallest population of 100 individuals. This proves that, PMX is the crossover operator which is affected from the data set growth for at least. As the population size grows, the gap between the execution times of small and larger data sets diminishes. As a result of this, for the larger populations, OX2 performs better than OX, PMX and OX'.

For each GA, there is a different trend of increase in runtime values with the increasing population sizes. The trends of increase in runtimes for OX, OX2, PMX and OX' for both training and control data are given in Appendix Q respectively.

According to the trend of increase graphs of runtime values, OX2 and PMX again outperform the other GAs because the trend of increase for LDST is similar to the trend of increase of SDST in both.

7.2 Genetic Operators of Successful Scenarios

In this study there is not only one solution of module range, but instead, there are many module ranges as the output because the software is run many times for different paramater combinations. All of these ranges are compared with that of the expert's suggestion and the aim is not only finding the best module range, but also determining the parameter combination giving the most number of reliable ranges.

When Table 5.2 is examined for the corresponding crate values of the successful scenarios, it can be observed that the number of scenarios having 0.75, 0.80 and 1.0 as the crate is more than the others for the smaller dataset as can be seen in Table 7.3. Figure 7.2 shows the performance of crate = 0.80 for SDST. For the larger dataset, higher crates produce more successful scenarios and the number of scenarios having 1.00 as the crate is more than the others.

Table 7.3 Number of successful scenarios for crate values

Dataset \ Crate	0.70	0.75	0.80	0.85	0.90	0.95	1.00
SDST	2	5	5	2	2	3	5
LDST	4	2	4	4	3	1	6

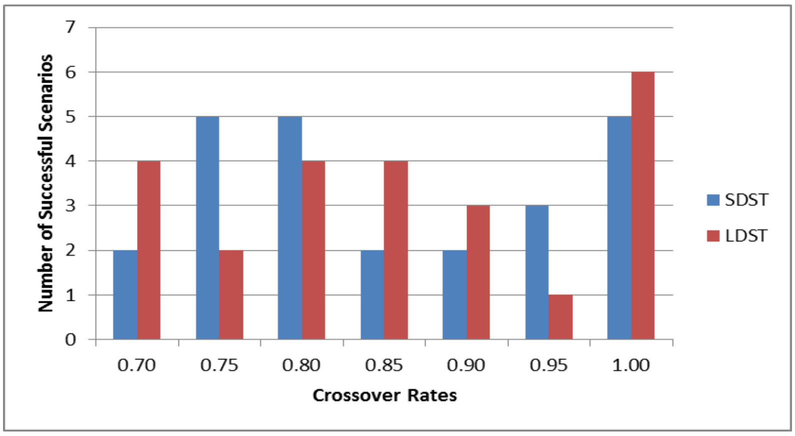


Figure 7.2 Performance of crossover rates in terms of the amounts of successful scenarios for SDST and LDST

The same extraction can be done for the mutation rates. The number of scenarios having 0.1 and 0.15 as the mrate is more than the others for the smaller dataset as can be seen in Table 7.4. Higher mrates does not perform well for a smaller dataset. For the larger dataset, higher mrates produce more successful scenarios and the number of scenarios having 0.2 as the mrate is more than the others.

Table 7.4 Number of successful scenarios for mate values.

Dataset \ Mrate	0.1	0.15	0.2
SDST	9	9	6
LDST	4	9	11

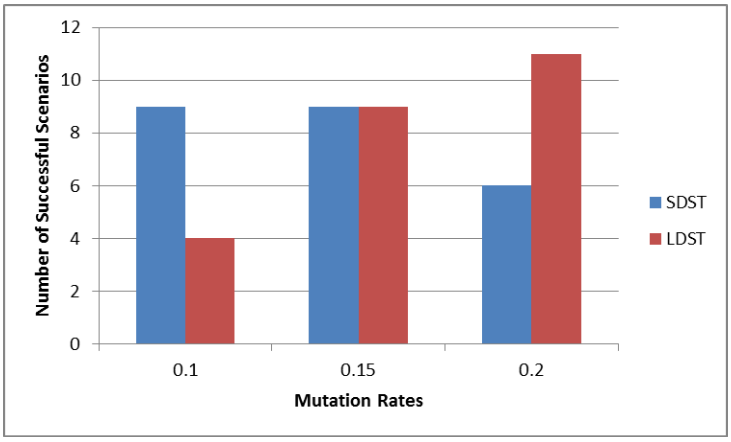


Figure 7.3 Performance of mutation rates in terms of the amounts of successful scenarios for SDST and LDST

When the scenarios giving the best module ranges for all GAs and population sizes for SDST and LDST of the control data set are examined, it is observed that the number of scenarios having 0.95 as the crate is more than the others for the smaller dataset as can be seen in Table 7.5. Figure 7.4 shows the performance of crate = 0.95 for SDST. For the larger dataset, lower crates produce more successful scenarios and the number of scenarios having 0.70 as the crate is more than the others.

Table 7.5 Number of successful scenarios for crate values

Dataset \ Crate	0.70	0.75	0.80	0.85	0.90	0.95	1.00
SDST	3	5	3	2	3	6	2
LDST	7	5	4	3	1	2	2

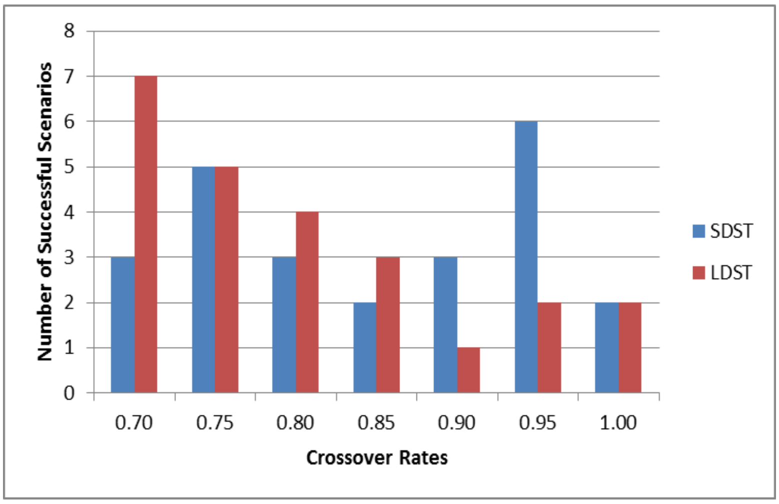


Figure 7.4 Performance of crossover rates in terms of the amounts of successful scenarios for SDST and LDST

The same extraction can be done for the mutation rates. The number of scenarios having 0.1 as the mrate is more than the others for the smaller dataset as can be seen in Table 7.6. Higher mrates does not perform well for a smaller dataset. For the larger dataset, the number of scenarios having 0.2 as the mrate is more than the others.

Table 7.6 Number of successful scenarios for mate values.

Dataset \ Mrate	0.1	0.15	0.2
SDST	12	5	7
LDST	10	2	12

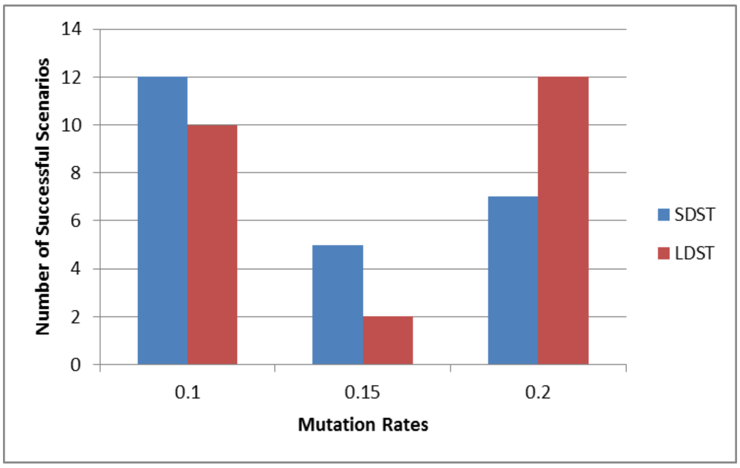
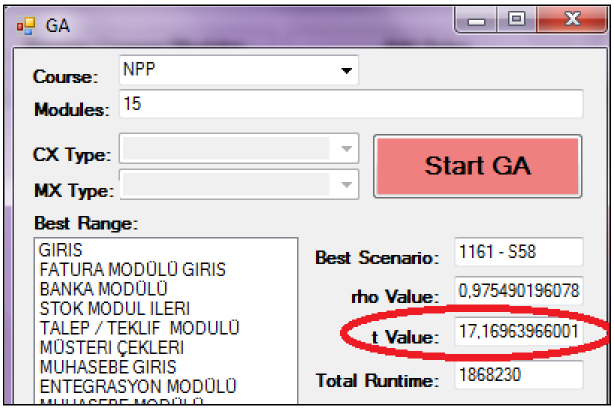


Figure 7.5 Performance of mutation rates in terms of the amounts of successful scenarios for SDST and LDST

7.3 Best t Values

The amount of successful solutions are calculated according to the t value formula of Spearman Rank Correlation (Formula (4) in Chapter Five). The calculated values are then compared with the corresponding value in t – Table (Appendix E). The corresponding t value of t – Table to compare is detected according to the Spearman Evaluation Criteria given in Table 5.5. The software implemented also gives the best t value of each 1260 runs and it can be seen on user screen as shown in Figure 7.6.



Course: NPP

Modules: 15

CX Type:

MX Type:

Best Range:

GIRIS
FATURA MODÜLÜ GIRIS
BANKA MODÜLÜ
STOK MODUL ILERI
TALEP / TEKLIF MODULÜ
MÜSTERI ÇEKLERI
MUHASEBE GIRIS
ENTEGRASYON MODÜLÜ

Best Scenario: 1161 - S58

rho Value: 0,975490196078

t Value: 17,16963966001

Total Runtime: 1868230

Figure 7.6 The User interface showing the best t value calculated. This user interface is an example of SDST for 100 individuals.

These best t value results for all population sizes for both SDST and LDST for the training and the control data are given in Table 7.7(a), Table 7.7(b), Table 7.7(c) and Table 7.7(d) respectively.

Table 7.7(a) Best t values for SDST for the training data set.

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	10.54	16.33	14.40	**17.17**
120	10.78	**17.17**	13.42	12.99
140	10.78	14.97	13.88	**19.26**
160	12.99	**18.13**	9.71	**18.13**
180	11.90	**22.33**	**22.33**	11.90
200	8.73	**20.63**	10.78	15.61

Table 7.7(b) Best t values for SDST for the control data set.

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	12.99	14.97	11.90	**18.13**
120	14.40	**19.26**	11.04	14.97
140	12.99	16.34	13.42	**20.63**
160	15.61	**16.34**	12.60	14.97
180	13.42	**22.33**	11.04	18.13
200	12.24	**14.97**	13.42	12.24

Table 7.7(c) Best t values for LDST for the training data set.

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	12.99	**20.85**	9.73	12.53
120	11.96	**17.81**	10.16	12.82
140	10.04	**21.35**	9.88	12.94
160	13.95	**17.61**	12.29	16.44
180	15.60	**15.82**	11.30	14.34
200	18.35	17.32	12.73	**21.02**

Table 7.7(d) Best t values for LDST for the control data set.

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	14.51	**22.06**	13.08	19.69
120	13.12	**20.25**	13.30	17.71
140	13.50	**20.86**	14.28	18.35
160	14.40	**18.13**	17.92	16.96
180	13.21	**23.93**	14.51	19.04
200	13.80	18.13	14.28	**22.83**

Having the highest t value indicates that, the result giving the highest t value is the best solution among the result set. Mostly, the best t values of different population sizes are from the operator OX2 for both the training and control data. This shows that 2 – point order crossover gives the best module range results predominantly. Complete t value results for SDST of 100 individuals for the training data are given in Appendix F.

7.4 Reliable Module Range Amounts for Each GA

Being the best solution is not the only anwser that this study looks for. Except the best result, number of all reliable results should be taken into consideration. For each GA and each population size, different amounts of reliable solutions have been acquired. The graphics of the number of reliable module ranges for each scenario in SDST and LDST for the training and control data are given in Appendix I.

The total number of reliable modules for each GA and population sizes are given in Table 7.8(a), Table 7.8(b), Table 7.8(c) and Table 7.8(d) for SDST and LDST for the training and the control data sets respectively.

Table 7.8(a) Total number of reliable results for SDST for the training data set

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	572	**947**	713	728
120	585	**999**	598	712
140	579	**1035**	637	766
160	624	**1049**	687	875
180	660	**1069**	687	848
200	667	**1124**	864	899

Table 7.8(b) Total number of reliable results for SDST for the control data set

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	967	**1148**	785	1045
120	991	**1188**	840	1097
140	1049	**1218**	914	1142
160	1053	**1231**	948	1180
180	1085	**1238**	986	1206
200	1114	**1253**	1050	1216

Table 7.8(c) Total number of reliable results for LDST for the training data set

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	836	**1181**	743	905
120	943	**1195**	964	1055
140	970	**1220**	1021	1150
160	1027	**1226**	1074	1162
180	1035	**1244**	1123	1205
200	1045	**1256**	1093	1197

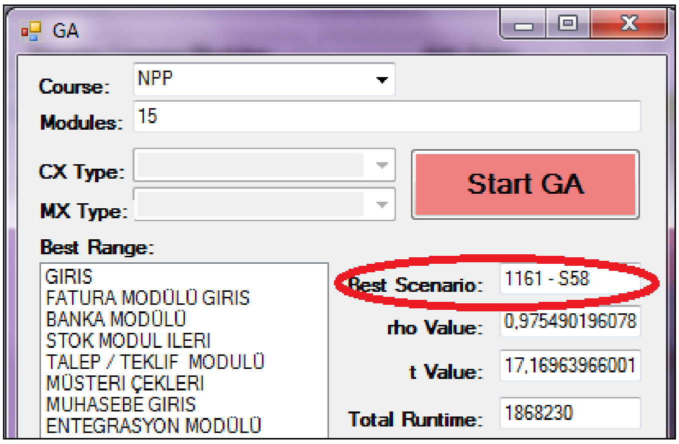
Table 7.8(d) Total number of reliable results for LDST for the control data set

	GAs			
Pop. Size	OX	OX2	PMX	OX’
100	1252	**1260**	1225	1258
120	1250	**1260**	1247	1258
140	1256	**1260**	1245	**1260**
160	**1260**	**1260**	1238	1259
180	1258	**1260**	1248	1259
200	1258	**1260**	1251	**1260**

According to the tables given above, OX2 is the best algorithm for both the training and the control data sets with highest numbers of reliable solutions among the other GAs. The graphics about the number of reliable solutions for each population size in both the training and control data are given in Appendix H.

7.5 Best Module Ranges

For each size of dataset, there are 6 best results for all sizes of population. The best reliable module range can be detected from the output shown in user interface. Figure 7.7 shows the best scenario number out of 1260.



Course: NPP

Modules: 15

CX Type:

MX Type:

Best Range:

GIRIS
FATURA MODÜLÜ GIRIS
BANKA MODÜLÜ
STOK MODUL ILERI
TALEP / TEKLIF MODULÜ
MÜSTERI ÇEKLERI
MUHASEBE GIRIS
ENTEGRASYON MODÜLÜ

Best Scenario: 1161 - S58

rho Value: 0,975490196078

t Value: 17,16963966001

Total Runtime: 1868230

Figure 7.7 The number of Best Scenario

The best reliable ranges for all dataset and population sizes are found among the raw .csv files. These module ranges are then compared with that of the expert’s suggestion. Best module ranges to be compared with the Expert's are given in Appendix J for the training data and in Appendix K for the control data respectively.

For the two different sizes of data sets, the human expert has two different suggestions. These suggestions are used to apply Spearman Rank Correlation on the calculated results to determine the reliability percentage of the solutions. Table 7.9(a) shows the reliability percentages of the module ranges of SDST for the training data and Table 7.9(b) the reliability percentages of the module ranges of SDST for the control data according to their population sizes when compared with the expert's suggestion (Wessa, 2012). The tables with the module ranges of given percentages are given in Appendix L for the training data and in Appendix M for the control data respectively in detail.

Table 7.9(a) Reliability Percentages for six population sizes of SDST for the training data set

	Reliability Percentages			
Pop. Size	OX	OX2	PMX	OX'
100	93.87%	97.30%	96.56%	**97.55%**
120	94.11%	**97.55%**	96.07%	95.83%
140	94.11%	96.81%	96.32%	**98.04%**
160	95.83%	**97.79%**	92.89%	**97.79%**
180	95.10%	**98.53%**	**98.53%**	95.10%
200	94.60%	**98.28%**	94.11%	97.06%

Table 7.9(b) Reliability Percentages for six population sizes of SDST for the control data set

	Reliability Percentages			
Pop. Size	OX	OX2	PMX	OX'
100	95.83%	96.81%	95.10%	**97.79%**
120	96.56%	**98.04%**	94.36%	96.81%
140	95.83%	97.30%	96.08%	**98.28%**
160	97.06%	**97.30%**	95.59%	96.81%
180	96.08%	**98.53%**	94.36%	97.80%
200	95.34%	**96.81%**	96.07%	95.34%

According to the results of SDST, population size of 180 with 2 – point order crossover and PMX give the most reliable results with a percentage of 98.53. The two – tailed p value for this range is less than 0.0001 ($p<0.01$); therefore this range is accepted as extremely significant. The crossover rates of the scenarios (S8 and S40 from Table 5.2.) giving the most reliable module range (crate) is 0.80 and 1.00. mutation rates are 0.10 and 0.15. choosing the crossover rate 0.80 or 1.00 and choosing the mutation rate 0.10 or 0.15 gives the optimum solutions having the highest reliability percentage.

Figure 7.8(a) and Figure 7.8(b) show the comparison graphics of most reliable results of SDST for training and control data with that of the expert's. In each case, the population size of 180 with OX2 give the best results for both training and control data. The x axis in figures represents the expected rank number of the module, where y axis represents the observed rank number of the module (Wessa, 2012). When the dots are ranged more linearly in the graphic, it indicates a more reliable range. The graphics of other population sizes for SDST for training and control data are given in Appendix N.

With 180 individuals, there exist two best results for OX2 and PMX in training data. Here the result of OX2 is considered because the execution time for OX2 is better.

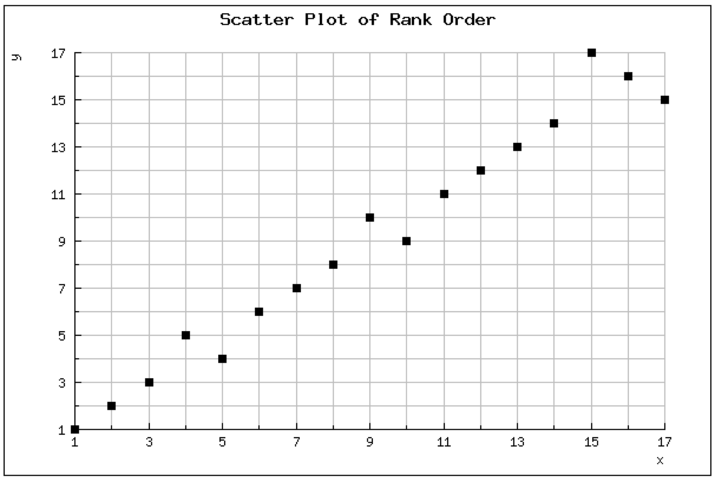


Figure 7.8(a) Comparing the most reliable module range result of training data – SDST for a population size of 180 with Expert's Suggestion. The solution is 98.53% reliable in OX2 and PMX.

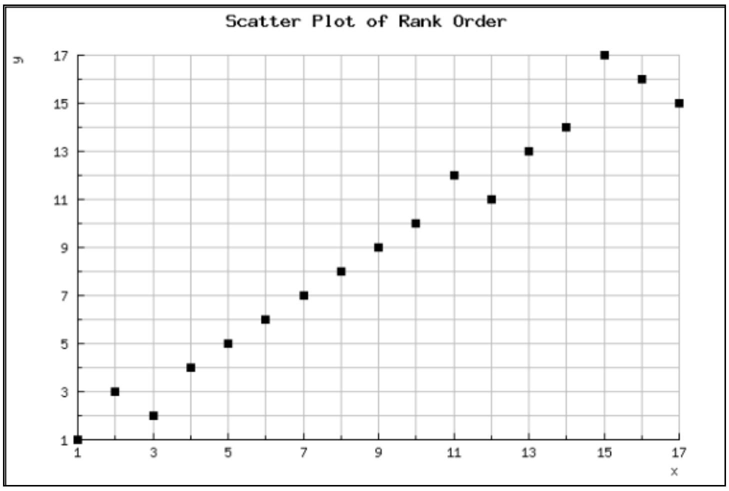


Figure 7.8(b) Comparing the most reliable module range result of control data – SDST for a population size of 180 with Expert's Suggestion. The solution is 98.53% reliable in OX2 and PMX.

Table 7.10(a) shows the reliability percentages of the module ranges of SDST for the training data and Table 7.10(b) the reliability percentages of the module ranges of LDST for the control data according to their population sizes when compared with the expert's suggestion (Wessa, 2012). The tables with the module ranges of given percentages are given in Appendix L for the training data and in Appendix M for the control data respectively in detail.

Table 7.10(a) Reliability Percentages for six population sizes of LDST for the training data set

	Reliability Percentages			
Pop. Size	OX	OX2	PMX	OX'
100	92.61%	**96.93%**	87.85%	92.12%
120	91.45%	**95.86%**	88.70%	92.43%
140	88.47%	**97.06%**	88.16%	92.57%
160	93.50%	**95.77%**	91.85%	95.19%
180	94.70%	**94.84%**	90.56%	93.81%
200	96.08%	95.64%	92.34%	**96.97%**

Table 7.10(b) Reliability Percentages for six population sizes of LDST for the control data set

	Reliability Percentages			
Pop. Size	OX	OX2	PMX	OX'
100	93.94%	**97.14%**	92.70%	96.57%
120	92.74%	**96.75%**	92.65%	95.81%
140	93.10%	**96.93%**	93.77%	96.08%
160	93.86%	**96.00%**	95.90%	95.46%
180	92.83%	**97.64%**	93.94%	96.35%
200	93.37%	95.99%	93.77%	**97.41%**

According to the results of LDST, population size of 140 with 2 – point order crossover gives the most reliable results with a percentage of 97.06. The two – tailed p value for this range is less than 0.0001 (p<0.01); therefore this range is accepted as extremely significant. The crossover rates of the scenario (S24 from Table 5.2.) giving the most reliable module range is 0.70. Mutation rate is 0.2. This means that, choosing the crossover rate 0.70 and choosing the mutation rate 0.2 gives the optimum solutions having the highest reliability percentage for a larger dataset with longer chromosomes.

Figure 7.9(a) and Figure 7.9(b) show the comparison graphics of most reliable results of LDST for training and control data with that of the expert's (Wessa, 2012). The population size of 140 for the training data and the population size of 180 for the control data give the best results. The graphics of other population sizes for LDST for training and control data are given in Appendix O. The deviations in the graphs depend on the characteristics of the data sets used for both trainnig and control data.

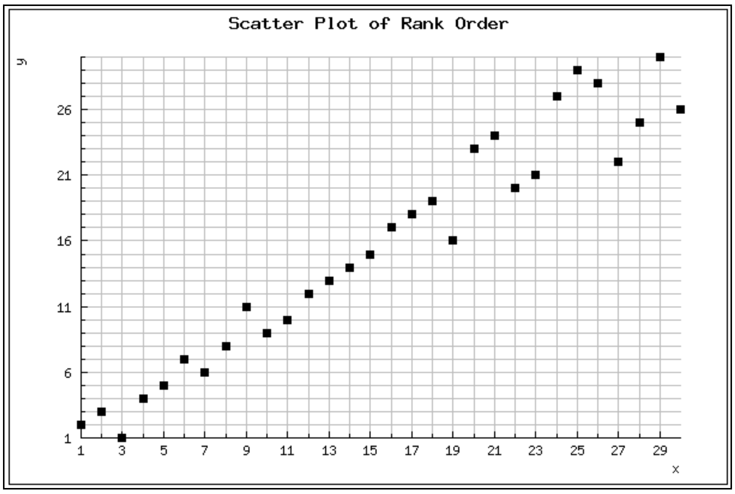


Figure 7.9(a) Comparing the most reliable module range result of training data – LDST for a population size of 140 with Expert's Suggestion. The solution is 97.06% reliable in OX2.

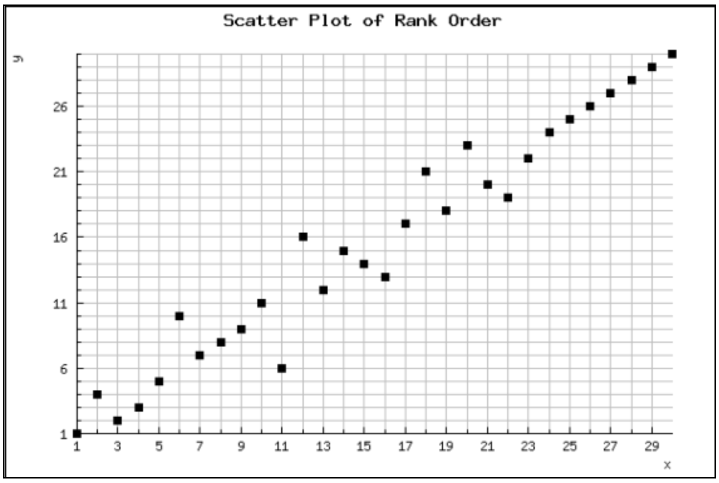


Figure 7.9(b) Comparing the most reliable module range result of control data – LDST for a population size of 180 with Expert’s Suggestion. The solution is 97.64% reliable in OX2.

7.6 Best Fitness Values

The fitness values for the best scenarios of the training and control data for SDST for all population sizes are given in Table 7.11(a) and Table 7.11(b). According to the best fitness values, SDST has many fitness values of 1 as maximum. This means that, a smaller data set has more chance to obtain the most successful module ranges.

Table 7.11(a) Best fitness values of training data for SDST.

Pop. Size	**OX**	**OX2**	**PMX**	**OX’**
100	1	1	1	1
120	0.990449	1	0.990449	1
140	1	1	0.977484	1
160	1	1	0.991783	0.990449
180	1	1	0.985465	0.991783
200	0.991783	1	1	1

Table 7.11(b) Best fitness values of control data for SDST.

Pop. Size	**OX**	**OX2**	**PMX**	**OX’**
100	1	0.970394	1	1
120	1	1	0.99279	0.988597
140	0.988597	1	1	1
160	1	1	0.99279	1
180	0.977452	1	0.991783	1
200	0.99279	1	1	1

The fitness values for the best scenarios of the training and control data for LDST for all population sizes are given in Table 7.12(a) and Table 7.12(b).

Table 7.12(a) Best fitness values of training data for LDST.

Pop. Size	**OX**	**OX2**	**PMX**	**OX'**
100	0,941969	0,989932	0,958366	0,940128
120	0,922741	0,975127	0,936834	0,974124
140	0,972276	0,992605	0,921092	0,961518
160	0,958085	0,943518	0,971504	0,977296
180	0,980838	0,990698	0,963411	0,989189
200	0,973544	0,993085	0,975832	0,984438

Table 7.12(b) Best fitness values of control data for LDST.

Pop. Size	**OX**	**OX2**	**PMX**	**OX'**
100	0.995902	0.995902	0.982835	0.995902
120	0.989461	0.995902	0.896098	0.983103
140	0.973344	0.995902	0.975586	0.995902
160	0.995902	0.995902	1	0.995902
180	0.989461	1	0.991837	0.995902
200	0.995902	1	0.981565	1

For LDST, the maximum fitness values are mostly obtained in OX2 (except for 160 individuals). This shows that, OX2 is more effective when the data set grows.

CHAPTER EIGHT
CONCLUSIONS

According to the preliminary results, it was concluded that the results of the tests that 2 – point order crossover operator is used give better results in terms of runtime and fitness values (Abidin and Çakır, 2011). Accordingly, the number of reliable module ranges found in tests with 2 – point order crossover operator was higher than the number of reliable module ranges with 1 – point order crossover operator. These tests were done for two population sizes (100 and 200).

In the final work, the population size interval was given as 100 – 200 (100, 120, 140, 160, 180 and 200). The system needed to test different sizes of populations because increasing the population size to a certain extend encourages the diversity of the population that the GA deals with. This means that, as the population size increases, the possibility of having individuals with various values of fitnesses also increases. With populations of small size, runtime and fitness values were more likely to be the best. As the population grows, finding "very reliable" solutions gets more difficult. Within this study, finding reliable solutons very similar to that of the expert's suggestions was not very difficult because the parameter tuning process helped to find the optimum module sequence with the most appropriate GA parameters.

It is very important to verify that the study stands as a reliable one among all other scientific researches in the relevant field literature. In this study, the output of the software was compared with the solution suggestion of a human to decide about the curriculum planning system's reliability. For this reason, all of the results obtained in the study with all scenarios and genetic operators were accepted as the best solutions at the beginning of the evaluation process and the Spearman Rank Correlation test was applied to all.

The Spearman test is a nonparametric statistical analysis method, which is used in cases where two data sets have to be observed with regard to their sequences. With this test, the module ranges similar to the range of the expert's suggestions and the numbers of reliable module ranges for each scenario are retained. Having many reliable module ranges is as important as obtaining the best module range because the system produces many reliable solutions and they can not be neglected during evaluation.

From the genetic operators point of view, keeping the crossover rates higher gives better results (0.9) both in 1 – point order and 2 – point order crossover operators. The mutation rates tested were extremely high in this study when compared with similar studies in literature, though the preliminary tests of the study indicated that lower mutation rates do not give better results in this kind of sequencing problems.

The way of representing the data became an important subject in the study. The platform independenceness of XML helped to store data in any format and it helped to convert the data to a mathematical representation like a matrix. Benefits of storing rules in matrix representation can be listed in a few topics: Convenience of storing and processing rules, easy transformation of rule set from an XML file to a sparse matrix, not taking up space in database. The source code to write the XML file with the prerequisite rules is given in Appendix G.

What was done with the dataset containing a curriculum content had to be verified with another dataset having the same characteristics; another curriculum for another course, if possible. To provide this, a control dataset, which had the same prerequisite features, was prepared and the same tests were done on it. The results verified that same parameter combinations gave the same successful module ranges in the same reliability percentages.

The results suggest that, the rule – based GA developed in this study can be used as a reliable system to optimize a curriculum sequence having tight prerequisite rules among the sections of educational material.

As the future work of this study, it is aimed to find a wider application area for the module ranking methodology in education. Since the presentation of the course content to students play an important role on their education, each course’s content should be ranked in the most appropriate order. This can be an important factor on raising more equipped students to compete with each other in industry after graduation.

Another future aim is to find the optimum curriculum plan with many lectures all together. The contents of the lectures can already be ranked according to the ranking method put forward in the thesis and the whole curriculum, e.g. eight semesters of courses for a faculty of four years, can also be ranked with this technique.

REFERENCES

Abdullah, S., Turabieh, H., McCollum, B. & McMullan, P. (2010a). A Multi – Objective Post Enrolment Course Timetabling Problems: A New Case Study. *Proceedings of IEEE Congress on Evolutionary Computation (CEC), IEEE,* 1 – 7. Retrieved August 1, 2011 from IEEE Database.

Abdullah, S., Turabieh, H., McCollum, B. & McMullan, P. (2010b). A Hybrid Metaheuristic Approach to the University Course Timetabling Problem. *Journal of Heuristics, 18 (1), Springer,* 1 - 23. Retrieved August 1, 2011 from Springer Database.

Abidin, D., Çakır, Ş. (2011). Rule-Based Genetic Algorithm for In-Service Training Curriculum Plan. *Proceedings of the International Conference on Computers, Digital Communications and Computing (ICDCCC'11)*, 160 – 166.

Abraham, A. (2005). Evolutionary Computation. In Sydenham P.H., & Thorn R., (Eds.). *Handbook of Measuring System Design* (910-911). John Wiley & Sons.

Adamuthe, A.C., Bichkar, R.S. (2011). Minimizing Job Completion Time in Grid Scheduling with Resource and Timing Constraints Using Genetic Algorithm. *Proceedings of the International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India*, 338 – 343.

Agustin – Blas, L.E., Salcedo – Sanz, S., Ortiz – Garcia, E.G., Portilla – Figueras, A., Perez – Bellido, A.M. (2009). A Hybrid Group Genetic Algorithm for Assigning Students to Preferred Laboratory Groups. *Expert Systems with Applications Vol. 36, Elsevier,* 10180 – 10187. Retrieved February 26, 2010 from Elsevier Database.

Ahmed, Z.H. (2010). Genetic Algorithm for the Traveling Salesman Problem Using Sequential Constructive Crossover Operator. *International Journal of Biometrics & Bioinformatics (IJBB), 3 (6),* 96 – 105. Retrieved April 4, 2012 from Computer Science Journals Database.

Ahmet, A., Zhoujun, L. (2010). Solving Course Timetabling Problem Using Interrelated Approach. *Proceedings of the International Conference on Granular Computing, IEEE*, 651 – 655. Retrieved August 1, 2011 from IEEExplore Database.

Aldasht, M., Alsaheb, M., Adi S., Qopita, M.A. (2009). University Course Scheduling Using Evolutionary Algorithms. *Proceedings of 4th International Multi – Conference on Computing in the Global Information Technology, IEEE,* 47 – 51. Retrieved August 1, 2011 from IEEE Database.

Alsmadi, O.MK., Abo-Hammour, Z.S., Abu-Al-Nadi, D.I., Algsoon, A. (2011). A Novel Genetic Algorithm Technique for Solving University Course Timetabling Problems. *Proceedings of 7th International Workshop on Systems, Signal Processing and Their Applications (WOSSPA'11), IEEE,* 195 – 198. Retrieved August 1, 2011 from IEEExplore Database.

Altay, A., Kayakutlu, G. & Topcu, Y.I. (2010). Win – Win Match Using a Genetic Algorithm. *Applied Mathematical Modelling 34, Elsevier,* 2749 – 2762. Retrieved August 1, 2011 from Elsevier Database.

Anghinolfi, D., Montemanni, R., Paolucci, M. & Gambardella, L.M. (2011). A Hybrid Particle Swarm Optimization Approach for the Sequential Ordering Problem. *Computers & Operations Research 38, Elsevier,* 1076 – 1085. Retrieved April 5, 2011 from Elsevier Database.

Ayob, M. & Jaradat, G. (2009). Hybrid Ant Colony Systems for Course Timetabling Problems. *Proceedings of IEEE Second Conference on Data Mining and Optimization, IEEE,* 120 – 126. Retrieved August 1, 2011 from IEEE Xplore Database.

Ballı, S., Karasulu, B., Uğur, A., Korukoğlu, S. (2009). Basketbolda Oyuncu Seçimi için Sinirsel-Bulanık Karar Destek Sistemi. *Journal of İTÜ/d, 8, No.1,* 15-25.

Bardadym, V. (1996). Computer-Aided School and University Timetabling: The New Wave. In E.K. Burke, P. Ross (Eds.). *Selected Papers from the 1st*

International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, vol. 1153, 22–45.

Barrero, D.F., Gonzales – Pardo, A., Camacho, D., R – Moreno, M.D. (2010). Distributed Parameter Tuning for Genetic Algorithms. *International Journal of Computer Science and Information Systems (ComSIS), Vol. 7. No. 3.*, 661 – 677. Retrieved March 7, 2012 from ComSIS Database.

Biroğul, S., Elmas, Ç., Çetin, A. (2011). Planning of the GSM Network Broadcast Control Channel with Data Fusion. *Expert Systems with Applications (38), Elsevier Ltd.*, 2421 – 2431. Retrieved August 1, 2011 from ScienceDirect Database.

Bissonette, V.L. (2011). *Critical Values of the t Distribution*. Retrieved January 23, 2012, from http://facultyweb.berry.edu/vbissonnette/tables/t.pdf.

Brizuela, C.A., Zhao, Y., Sannomiya, N. (2001). No – Wait and Blocking Job – Shops: Challenging Problems for GA's. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Vol. 4*, 2349 – 2354. Retrieved March 8, 2012 from IEEE Database.

Burke, E.K., De Causmaecker, P., Berghe, G.V., Van Landeghem, H. (2004). The State of the Art of Nurse Rostering. *Journal of Scheduling 7 (6)*, 441–499.

Burke, E.K., Jackson, S., Kingston, H., Weare, F. (1997). Automated University Timetabling: The State of the Art. *The Computer Journal 40 (9)*, 565–571.

Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R. (2007). A Graph-Based Hyper-Heuristic for Educational Timetabling Problems. *European Journal of Operational Research (176)*, 177-192.

Burke, E.K., Petrovic, S. (2002). Recent Research Directions in Automated Timetabling. *European Journal of Operational Research 140 (2)*, 266–280.

Cardoen, B., Demeulemeester, E., Belien, J. (2010). Operating Room Planning and Scheduling: A Literature Review. *European Journal of Operational Research, Vol. 201*, *Elsevier*, 921-932. Retrieved September 27, 2010 from ScienceDirect Database.

Carter, M., Laporte, G. (1996). Recent Developments in Practical Exam Timetabling. In E.K. Burke, P. Ross (Eds.). *Selected Papers from the 1st International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, vol. 1153*, 3–21.

Carter, M., Laporte, G. (1998). Recent Developments in Practical Course Timetabling. In: E.K. Burke, P. Ross (Eds.). *Selected Papers from the 2nd International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science, vol. 1408,* 3–19.

Castelli, M., Manzoni, L., Vanneschi, L. (2011). The Effect of Selection from Old Populations in Genetic Algorithms. *Proceedings of Genetic and Evolutionary Computation Conference GECCO’11*, 161 – 162. Dublin, Ireland.

Castillo, O., Melin, P. (1996). Automated Mathematical Modelling for Financial Time Series Prediction Using Fuzzy Logic, Dynamical System Theory and Fractal Theory. *Proceedings of Computational Intelligence for Financial Engineering IEEE/IAFE’96*, 120 – 126. NY: IEEE Press. Retrieved September 20, 2007 from IEEExplore Database.

Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W. (2008). Soft Computing for Hybrid Intelligent Systems. *Studies in Computational Intelligence 154*, 387 – 388. Springer – Verlag.

Chakraborty, M., Chakraborty, U.K. (1997). An Analysis of Linear Ranking and Binary Tournament Selection in Genetic Algorithms. *Proceedings of International Conference on Information, Communications and Signal*

Processing ICICS’97, Singapore, 9 – 12 September, 1997, 407 – 411. Retrieved January 5, 2010 from IEEE Xplore Database.

Chakravorty, S., Thukral, M. (2009). Choosing Distribution Substation Location Using Soft Computing Technique. *Proceedings of International Conference on Advances in Computing, Communication and Control (ICAC3’09), ACM*, 53-55.

Chan F.T.S., Choy K.L., Bibhushan. (2011). A Genetic Algorithm – Based Scheduler for Multiproduct Parallel Machine Sheet Metal Job Shop. *Expert Systems with Applications, Elsevier, Vol. 38,* 8703 – 8715. Retrieved August 5, 2011 from Elsevier Database.

Chang, P.-C., Huang, W.-H., Ting, C.-J. (2010). Dynamic Diversity Control in Genetic Algorithm for Mining Unsearched Solution Space in TSP Problems. *Expert Systems with Applications Vol. 37, Elsevier,* 1863 – 1878. Retrieved February 26, 2010 from Elsevier Database.

Chang, T.-J., Yang, S.-C., Chang, K.-J. (2009). Portfolio Optimization Problems in Different Risk Measures Using Genetic Algorithm. *International Journal of Expert Systems with Applications 36*, 10529 – 10537. Retrieved July 6, 2009 from ScienceDirect Database.

Chaoan, L. (2007). The Expert System of Product Design Based on CBR and GA. *International Conference on Computational Intelligence and Security Workshops, IEEE,* 144-147. Retrieved June 30, 2008 from IEEExplore Database.

Cheang, B., Li, H., Lim, A., Rodrigues, B. (2003). Nurse Rostering Problems: A Bibliographic Survey. *European Journal of Operational Research 151 (3)*, 447–460.

Cheng, W., Shi, H., Yin, X., Li, D. (2011). An Elitism Strategy Based Genetic Algorithm for Streaming Pattern Discovery in Wireless Sensor Networks. *IEEE Communication Letters, Vol. 15, No. 4*, 419 – 421. Retrieved January 2, 2012 from IEEExplore Database.

Chinnasri, W. & Sureerattanan, N. (2010). Comparison of Performance between Different Selection Strategies on Genetic Algorithm with Course Timetabling Problem. *Proceedings of IEEE International Conference on Advanced Management Science (ICAMS) 2,* 105 – 108. Retrieved August 1, 2011 from IEEE Xplore Database.

Chiou, C.-W., Wu, M.-C. (2009). A GA-Tabu Algorithm for Scheduling In-Line Steppers in Low-Yield Scenarios. *International Journal of Expert Systems with Applications 36*, 11925 – 11933. Retrieved July 6, 2009 from ScienceDirect Database.

Chou, J-S. (2009). Generalized Linear Model-Based Expert System for Estimating the Cost of Transportation Projects. *Expert Systems with Applications 36, Elsevier*, 4253-4267. Retrieved February 26, 2010 from ScienceDirect Database.

Choy, K.L., Leung, Y.K., Chow, H.K.H., Poon, T.C., Kwong, C.K., Ho, G.T.S., et al. (2011). A Hybrid Scheduling Decision Support Model for Minimizing Job Tardiness in a Make-to-Order Based Mould Manufacturing Environment. *Expert Systems with Applications (38), Elsevier Ltd.,* 1931 – 1941. Retrieved August 1, 2011 from ScienceDirect Database.

Chung, J.-W., Oh, S.-M., Choi, I.-C. (2009). A hybrid Genetic Algorithm for Train Sequencing in the Korean Railway. *The International Journal of Management Science, Omega 37,* 555 – 565. Elsevier Ltd. Retrieved August 5, 2011 from ScienceDirect Database.

Coley, D.A. (1998). An Introduction to Genetic Algorithms for Scientists and Engineers. *World Scientific, London*. Retrieved September 17, 2012 from http://read.pudn.com.

Cupic, M., Golub, M. & Jakobovic, D. (2009). Exam Timetabling Using Genetic Algorithm. *Proceedings of the International Conference on Information Technology Interfaces (ITI'09),* 357 – 362. Cavtat, Croatia. Retrieved August 1, 2011 from Springer Database.

Davis, L. (1991). Handbook of Genetic Algorithms. *Van Nostrand Reinhold, New York.*

De Marcos, L., Barchino, R., Martinez, J.J. (2008). Evolutionary Approaches for Curriculum Sequencing. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education ITiCSE'08,* 373. Madrid, Spain. Retrieved January 5, 2009 from ACM Digital Library Database.

De Marcos, L., Martinez, J.-J., Gutierrez, J.-A., Barchino, R., Gutierrez, J.-M. (2008). An Evolutionary Approach for Competency – Based Curriculum Sequencing. *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO'08,* 1697 – 1698. Atlanta, Georgia, USA. Retrieved January 5, 2009 from ACM Digital Library Database.

De Marcos, L., Barchino, R., Martinez, J.-J., Gutierrez, J.-A., Hilera, J.-R. (2008). Competency-Based Curriculum Sequencing: Comparing Two Evolutionary Approaches. *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,* 339-342. Retrieved August 6, 2009 from IEEExplore Database.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms.* (103-104). England: John Wiley and Sons LTD.

Deep, K. & Mebrahtu, H. (2011). New Variations of Order Crossover for Travelling Salesman Problem. *International Journal of Combinatorial Optimization and Informatics 2 (1),* 2 – 13. Retrieved April 4, 2012 from IJCOPI Database.

Derakhshi, M.R.F., Zandi, M. (2010). A Solution to the Problem of University Examinations Timetabling Through Information Related to the Units Taken Previous Terms. *Proceedings of 2nd International Conference on Computer and Automation Engineering (ICCAE'10), Vol. 1,* 559 – 562. IEEE. Retrieved August 1, 2011 from IEEExplore Database.

Easton, K., Nemhauser, G., & Trick, M. (2004). Sports Scheduling. In J. Leung, (Ed.). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. (Chapter 52). CRC Press.

Ebrahimipour, V., Nagasaka, K. (2003). Deterministic and Non-Deterministic Methods for Power Sector Performance Ranking Based on Machine Productivity. *Proceedings of International Conference of IEEE Power Tech Vol. 3, Bologna, Italy,* 8. Retrieved December 16, 2011 from IEEExplore Database.

Eiben, A.E., Smit, S.K. (2011). Parameter Tuning for Configuring and Analyzing Evolutionary Algorithms. *Journal of Swarm and Evolutionary Computation Vol. 1, Issue. 1, Elsevier,* 19 – 31.

Elmas, Ç. (2007). *Yapay Zeka Uygulamaları* (388-402). Ankara: Seçkin Yayınevi.

Fakhrzad, M.B., Zare, H.K. (2009). Combination of Genetic Algorithm with Lagrange Multipliers for Lot – Size Determination in Multi – Stage Production Scheduling Problems. *Expert Systems with Applications Vol. 36, Elsevier,* 10180 – 10187. Retrieved February 26, 2010 from Elsevier Database.

Fan, Y.-N., Tseng, T.-L., Chern, C.-C., Huang, C.-C. (2009). Rule Induction Based on an Incremental Rough Set. *International Journal of Expert Systems with Applications 36*, 11439 – 11450. Retrieved July 6, 2009 from ScienceDirect Database.

Fernandez – Prieto, J.A., Canada – Bago, J., Gadeo – Martos, M.A. & Velasco, J.R. (2011). Optimisation of Control Parameters for Genetic Algorithms to Test Computer Networks Under Realistic Traffic Loads. *Journal of Applied Soft Computing, 11 (4), Elsevier,* 3744 – 3752. Retrieved April 5, 2011 from Elsevier Database.

Fernandez, A., del Jesus, M.J., Herrera, F. (2009). Hierarchical Fuzzy Rule Based Classification Systems with Genetic Rule Selection for Imbalanced Data – Sets.

International Journal of Approximate Reasoning, Vol 50, Elsevier, 561 – 577. Retrieved January 5, 2010 from IEEE Xplore Database.

Garcia-Camacho, F., Gallardo-Rodriguez, J.J., Sanchez-Miron, A., Christi, Y., Molina-Grima, E. (2011). Genetic Algorithm Based Medium Optimization for a Toxic Dinoflagellate Microalga. *Harmful Algae (10), Elsevier*, 697 – 701. Retrieved December 16, 2011 from ScienceDirect Database.

Giarratano, J., & Riley, G. (2004). *Expert Systems: Principles and Programming* (4th Ed.). Course Technology Ptr.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA: Addison Wesley.

Greffenstette, J.J., Baker, J.E. (1989). How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. *Proceedings of the 3rd International Conference on Genetic Algorithms*, 20 – 27. San Mateo, CA: Morgan Kaufmann Publishers.

Hamid, S., Mirhosseyni, L., Webb, P. (2009). A Hybrid Fuzzy Knowledge-Based Expert System and Genetic Algorithm for Efficient Selection and Assignment of Material Handling Equipment. *International Journal of Expert Systems with Applications 36*, 11875 – 11887. Retrieved July 6, 2009 from ScienceDirect Database.

He, Y., Hui, C. – W. (2008). A Rule – Based Genetic Algorithm for the Scheduling of Single – Stage Multi – Product Batch Plants with Parallel Units. *Computers and Chemical Engineering Vol. 32,* 3067 – 3083. Retrieved February 26, 2010 from Elsevier Database.

Holland, H.J. (1975). *Adaptation in Natural and Artificial Systems.* Cambridge, MIT Press.

Hong, C.-M., Chen, C.-M., Chang, M.-H., Chen, S.-C. (2007). Intelligent Web-Based Tutoring System with Personalized Learning Path Guidance. *Proceedings* *of 7th IEEE International Conference on Advanced Learning Technologies (ICALT2007),* 512-516. Retrieved June 30, 2008 from IEEExplore Database.

Hoste, K., Phansalkar, A., Eeckhout, L. (2006). Performance Prediction Based on Inherent Program Similarity. *Proceedings of the 15th International Conference on Parallel Architectures and Compilation Techniques, PACT'06,* 114 – 122. Washington: Seattle, USA. Retrieved December 16, 2011 from ACM Digital Library Database.

Hu, X.-B., DiPaolo, E.A. (2011). A Ripple – Spreading Genetic Algorithm for the Aircraft Sequencing Problem. *Journal of Evolutionary Computation, Vol. 19(1), Massachusetts Institute of Technology,* 77 – 106. Retrieved August 5, 2011 from MIT Press Journals Database.

Hung, J.-C. (2009). A Fuzzy GARCH Model Applied to Stack Market Scenario Using a Genetic Algorithm. *Expert Systems with Applications Vol. 36, Elsevier,* 11710 – 11717. Retrieved July 6, 2009 from Elsevier Database.

Jat, S.N. & Yang, S. (2011). A Hybrid Genetic Algorithm and Tabu Search Approach for Post Enrolment Course Timetabling. *Journal of Scheduling 14 (6), Springer,* 617 – 637. Retrieved August 1, 2011 from Springer Database.

Jorge, A. S.A., Martin, C., Hugo, T.-M. (2010). Several Strategies to Improve the Performance of Hyperheuristics for Academic Timetabling Design Problem. *Proceedings of the IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA'10),* 102 – 107. Retrieved August 1, 2011 from IEEExplore Database.

Jorge, A., S.A., Martin, C., Hector, P. (2010). A New Approach of Design for the Academic Timetabling Problem through Genetic Algorithms. *Proceedings of the IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA'10),* 96 – 101. IEEE. Retrieved August 1, 2011 from IEEExplore Database.

Juang, Y. – S., Lin, S. – S., Kao, H. – P. (2007). An Adaptive Scheduling System with Genetic Algorithms for Arranging Employee Training Programs. *Expert Systems with Applications Vol. 33, Elsevier,* 642 – 651. Retrieved February 26, 2010 from Elsevier Database.

Julstrom, B.A. (2004). Encoding Bounded – Diameter Spanning Trees with Permutations and with Random Keys. *Proceedings of 6th Annual Conference on Genetic and Evolutionary Computation Conference GECCO'04*, 1272 – 1281. Seattle, Washington, USA. Retrieved April 5, 2011 from Elsevier Database.

Julstrom, B.A. (2009). String and Permutation Coded Genetic Algorithms for the Static Weapon – Target Assignment Problem. *Proceedings of 11th Annual Conference on Genetic and Evolutionary Computation Conference GECCO'09*, 2553 – 2558. Montreal, Quebec, Canada. Retrieved August 5, 2011 from ACM Digital Library Database.

Karol, B., Tomasz, B., Henryk, K. (2006). Parallelisation of Genetic Algorithms for Solving University Timetabling Problems. *Proceedings of the International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06), IEEE,* 325 – 330. Retrieved January 5, 2010 from IEEE Xplore Database.

Kaya, M. (2010). Autonomous Classifiers with Understandable Rule Using Multi – Objective Genetic Algorithms. *Expert Systems with Applications Vol. 37, Elsevier,* 3489 – 3494. Retrieved February 26, 2010 from Elsevier Database.

Kaya, İ. (2009). A Genetic Algorithm Approach to Determine the Sample Size for Control Charts with Variables and Attributes. *International Journal of Expert Systems with Applications 36*, 8719 – 8734. Retrieved July 6, 2009 from ScienceDirect Database.

Khonggamnerd, P. & Innet, S. (2009). On Improvement of Effectiveness in Automatic University Timetabling Arrangement with Applied Genetic

Algorithm. *Proceedings of IEEE 4th International Conference on Computer Sciences and Convergence Information Technology,* 1266 – 1270. Retrieved August 1, 2011 from IEEE Xplore Database.

Kim, J.-S. (2007). Development of a User-Friendly Expert System for Composite Laminate Design. *Composite Structures 79, Elsevier*, 76-83. Retrieved October 5, 2007 from ScienceDirect Database.

Kuroki, Y., Young, G.S., Haupt, S.E. (2010). UAV Navigation by an Expert System for Contaminant Mapping with a Genetic Algorithm. *Expert Systems with Applications, Vol. 37, Issue 6*, 4687 - 4697. Retrieved February 26, 2010 from ScienceDirect Database.

Kwan, R. (2004). Bus and Train Driver Scheduling. In: J. Leung (Ed.). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis* (Chapter 51). CRC Press.

Lau, H.C.W., Tang, C.X.H., Ho, G.T.S., Chan, T.M. (2009). A Fuzzy Genetic Algorithm fort he Discovery of Process Parameter Settings Using Knowledge Representation. *International Journal of Expert Systems with Applications 36,* 7964 – 7974. Retrieved July 6, 2009 from ScienceDirect Database.

Lee, J. – Y., Kim, M. – S. & Lee, J. – J. (2011). Compact Genetic Algorithms Using Belief Vectors. *Applied Soft Computing 11 (4), Elsevier,* 3385 – 3401. Retrieved April 5, 2011 from Elsevier Database.

Lee, W. – C., Wu, C. – C., Liu, M. – F. (2009). A Single – Machine Bi – Criterion Learning Scheduling Problem with Release Times. *Expert Systems with Applications Vol. 36, Elsevier,* 10295 – 10303. Retrieved February 26, 2010 from Elsevier Database.

Lee, Y.-G., McKay, B. (2011). How Hard Should We Run? Trading off Exploration for Dynamic Optimization Problems. *Proceedings of 11th Annual Conference on*

Genetic and Evolutionary Computation Conference GECCO'13, 1187 – 1194. Dublin, Ireland. Retrieved January 2, 2012 from ACM Digital Library Database.

Li, X., Lv, X., Mei, W., Xu, H. (2010). Algorithm for Solving Time Table Questions Based on GA. *Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences (ICIS)*, 18 – 21.

Li, Y., Zhang, S., Zeng, X. (2009). Research of Multi-Population Agent Genetic Algorithm for Feature Selection. *International Journal of Expert Systems with Applications 36*, 11570 – 11581. Retrieved July 6, 2009 from ScienceDirect Database.

Li-Li, L., Ding-Wei, W. (2008). Permutation-Based Dual Genetic Algorithm Applied in Dynamic Sequencing Optimizations. *Journal of Systems Engineering – Theory and Practice (SETP), Vol.28, Issue 11,* 129 – 134. Retrieved August 5, 2011 from ScienceDirect Database.

Lin, J.-L., Wei, M.-C. (2009). Genetic Algorithm-Based Clustering Approach for k-Anonymization. *International Journal of Expert Systems with Applications 36,* 9784 – 9792. Retrieved July 6, 2009 from ScienceDirect Database.

Liu, M., Sun, Z.-J., Yan, J.-W., Kang, J.-S. (2011). An Adaptive Annealing Genetic Algorithm for the Job – Shop Planning. *Expert Systems with Applications, Elsevier, Vol. 38,* 9248 – 9255. Retrieved August 5, 2011 from Elsevier Database.

Liu, S.-H., Mernik, M., Bryant, B.R. (2004). Parameter Control in Evolutionary Algorithms by Domain – Specific Scripting Language PPC$_{EA}$. *Proceedings of International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2004. Ljubljana, Slovenia,* 41 – 50. Retrieved March 7, 2012 from http://bioma.ijs.si/.

Lochtefeld, D.F. & Ciarallo, F.W. (2010). Deterministic Helper – Objective Sequences Applied to Job – Shop Scheduling. *Proceedings of 12th Annual Conference on Genetic and Evolutionary Computation Conference GECCO'10,* 431 – 438. Portland, Oregon, USA. Retrieved January 2, 2012 from ACM Digital Library Database.

Marano, G.C. (2011). Modified Genetic Algorithm for the Dynamic Identification of Structural Systems Using Incomplete Measurements. *Journal of Computer – Aided Civil and Infrastructure Engineering (26)*, 92 – 110.

Maulik, U., Bandyopadhyay, S., & Mukhopadhyay, A. (2011). *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*. (31). Berlin: Springer – Verlag.

Mendes, J.M., (September 23-25, 2008). Project Scheduling Using a Competitive Genetic Algorithm. *Proceedings of 9th WSEAS International Conference on Simulation, Modeling and Optimization (SMO'08)*, 39 – 42. Santander, Cantabria, Spain.

Meng, X., Zhang, P., Li, C. (2010). Aircraft Category Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling. *Proceedings of the 29th IEEE Chinese Control Conference,* 5188 – 5193. Beijing, China. Retrieved August 5, 2011 from IEEE Database.

Mishra, A., Jhapate, A.K., Kumar, P. (2009). Desiging Rule Base for Genetic Feedback AlgorithmBased Network Security Policy Framework Using State Machine. *Proceedings of International Conference on Signal Processing Systems,* 415 – 417. IEEE. Retrieved January 2, 2012 from IEEExplore Database.

Moghaddam, M.E., Monyadi, M.R. (2011). An Immune – Based Genetic Algorithm with Reduced Search Space Coding for Multiprocessor Task Scheduling Problem. *International Journal of Parallel Programming, 2,* 225 – 257. Retrieved August 5, 2011 from SpringerLink Database.

Mokhtari, H., Abadi, I.N.K., Zegordi, S.H. (2011). Production Capacity Planning and Scheduling in a No-Wait Environment with Controllable Processing Times: An Integrated Modelling Approach. *Expert Systems with Applications (38), Elsevier Ltd.,* 12630 – 12642. Retrieved August 1, 2011 from ScienceDirect Database.

Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R. (2011). Solving a Capacitated Fixed-Charge Transportation Problem by Artificial Immune and Genetic Algorithms with Prüfer Number Representation. *Expert Systems with Applications (38), Elsevier Ltd.,* 10462 – 10474. Retrieved August 1, 2011 from ScienceDirect Database.

Naznin, F., Sarker, N., Essam, D., (2010). DGA: Decomposition with Genetic Algorithm for Multiple Sequence Alignment. *Proceedings of IEEE Symposium on Computational Intelligence, Bioinformatics and Computational Biology (CIBCB'10),* 1 – 8. Retrieved August 5, 2011 from IEEE Database.

Nguyen, H.H., Uraikul, V., Chan, C.W., Tontiwachwuthikul, P. (2008). A Comparison of Automation Techniques for Optimization of Compressor Scheduling. *Advances in Engineering Software 39*, 178-188.

Olsen, A.L. (2009). Evolutionary Computation in the Undergraduate Curriculum. *Journal of Computing Sciences in Colleges, Vol. 25, Issue 2,* 115 – 121. Retrieved February 26, 2010 from ACM Digital Library Database.

Otero, J., Sanchez, L., Alcala-Fdez, J. (2008). Fuzzy-Genetic Optimization of the Parameters of a Low Cost System for the Optical Measurement of Several Dimensions of Vehicles. *Journal of Soft Computing 12*, 751 – 764. Retrieved July 16, 2009 from Web of Science Database.

Pan, J.C.-H., Huang, H.C. (2009). A Hybrid Genetic Algorithm for No – Wait Job Shop Scheduling Problems. *Expert Systems with Applications Vol. 36, Elsevier,* 5800 – 5806. Retrieved February 26, 2010 from Elsevier Database.

Pavan, M., Mauri, A., Todeschini, R. (2004). Total Ranking Models by the Genetic Algorithm Variable Subset Selection (GA-VSS) Approach for Environmental Priority Settings. *Analytical and Bioanalytical Chemistry, 380 (3)*, 430 – 444. Springer-Verlag.

Petrovic, S., & Burke, E.K. (2004). University Timetabling. In J. Leung, (Ed.), *Handbook of Scheduling: Algorithms, Models, and Performance Analysis* (Chapter 45). CRC Press.

Picek, S., Golub, M. (2010). On the Efficiency of Crossover Operators in Genetic Algorithms with Binary Representation. *Proceedings of the 11th WSEAS International Conference on Neural Networks, 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems (NN'10/EC'10/FS'10)*, 167-172.

Pillay, N., Banzhaf, W. (2010). An Informed Genetic Algorithm for the Examination Timetabling Problem. *Applied Soft Computing Vol. 10, Elsevier,* 457 – 467. Retrieved February 26, 2010 from Elsevier Database.

Pop, P.C. & Iordache, S. (2011). A Hybrid Heuristic Approach for Solving the Generalized Travelling Salesman Problem. *Proceedings of Genetic and Evolutionary Computation Conference GECCO’11,* 481 – 488. Dublin, Ireland. Retrieved September 17, 2012 from ACM Digital Library Database.

Prieto, J.A.F., Perez, J.R.V. (2008). Adaptive Genetic Algorithm Control Parameter Optimization to Verify the Network Protocol Performance. *Proceedings of 12th International Conference on Information Processing and Management of Uncertainty in Knowledge – Based Systems, Malaga, Spain,* 785 – 791. Retrieved March 7, 2012 from http://www.gimac.uma.es/IPMU08/proceedings/.

Raghavjee, R. & Pillay, N. (2010). Using Genetic Algorithms to Solve the South African School Timetabling Problem. *Proceedings of IEEE Second World Congress on Nature and Biologically Inspired Computing (NaBIC), IEEE,* 286 – 292. Retrieved August 1, 2011 from IEEE Xplore Database.

Raiford, D.W., Krane, D.E., Doom, T.E.W., Raymer, M.L. (2011). A Genetic Optimization Approach for Isolating Translational Efficiency Bias. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 8, No.2,* 342 – 352. Retrieved December 16, 2011 from ACM Digital Library Database.

Ramteke, M., Srinivasan, R. (2011). Novel Genetic Algorithm for Short Term Scheduling of Sequence Dependent Changeovers in Multiproduct Polymer Plants. *The Journal of Computers and Chemical Engineering. Vol. 35, Issue 12,* 2945 – 2959. Retrieved August 5, 2011 from ScienceDirect Database.

Rau, H., Cho, K.-H. (2009). Genetic Algorithm Modeling for the Inspection Allocation in Reentrant Production Systems. *International Journal of Expert Systems with Applications 36*, 11287 – 11295. Retrieved July 6, 2009 from ScienceDirect Database.

Rodriguez, M.A., Escalante, D.M., Peregrin A. (2011). Efficient Distributed Genetic Algorithm for Rule Extraction. *Journal of Applied Soft Computing Vol. 11, Issue 1,* 733 – 743. Retrieved March 8, 2012 from IEEE Database.

Roeva, O. (May 2-4, 2008). Improvement of Genetic Algorithm Performance for Identification of Cultivation Process Models. *Proceedings of 9th WSEAS International Conference on Evolutionary Computing (EC'08),* 34 – 39. Sofia, Bulgaria.

Roland, B., Di Martinelly, C., Riane, F., Pochet, Y. (2010). Scheduling an Operating Theatre Under Human Resource Constraints. *Computers and Industrial Engineering Vol. 58, Issue 2,* 212 – 220. Retrieved February 26, 2010 from Elsevier Database.

Rom, W.O., Slotnick, S.A. (2009). Order Acceptance Using Genetic Algorithms. *Computers & Operations Research Vol. 36,* 1758 – 1767. Elsevier Ltd. Retrieved August 5, 2011 from ScienceDirect Database.

Sasikala, J., Ramaswamy, M. (2010). Optimal Gamma Based Fixed Head Hydrothermal Scheduling Using Genetic Algorithm. *Expert Systems with Applications Vol. 37, Elsevier,* 3352 – 3357. Retrieved February 26, 2010 from Elsevier Database.

Schaerf, A. (1999). A Survey of Automated Timetabling. *Artificial Intelligence Review 13 (2),* 87–127.

Selouani, S.-A. (2011). *Speech Processing and Soft Computing.* New York: Springer.

Sheskin, D.J. (2000). *Handbook of Parametric and Non-Parametric Statistical Procedures* (2nd Ed.). (870 - 872). Chapman & Hall / CRC.

Shu, Q., Gong, G., Wang, J. (2010). STEP – NC Based Machining Sequencing Using Genetic Algorithm. *Proceedings of Chinese Control and Decision Conference (CCDC), IEEE,* 1299 – 1304. Retrieved August 5, 2011 from IEEE Database.

Shukla, A., Tiwari, R., & Kala R., (2010). *Real Life Applications of Soft Computing.* (4-8). CRC Press by Taylor and Francis Group LLC.

Singh, A., Baghel, A.S. (2009). A New Grouping Genetic Algorithm Approach to the Multiple Traveling Salesperson Problem. *Journal of Soft Computing 13,* 95 – 101. Retrieved July 16, 2009 from Web of Science Database.

Sivanandam, S.N., & Deepa, S.N. (2008). *Introduction to Genetic Algorithms*. (49). Berlin: Springer – Verlag.

Smit, S.K., Eiben, A.E. (2010). Parameter Tuning of Evolutionary Algorithms: Generalist and Specialist. *In Applications of Evolutionary Computation Vol. 6024,* 542 – 551. Retrieved August 5, 2011 from Springer Database.

Souai, N., Teghem, J. (2009). Genetic Algorithm Based Approach for the Integrated Airline Crew – Pairing and Rostering Problem. *European Journal of Operational*

Research, Vol. 199, Elsevier, 674 – 683. Retrieved February 26, 2010 from Elsevier Database.

Srndic, N., Pandzo, E., Dervisevic, M., Konjicija, S. (2009). The Application of a Parallel Genetic Algorithm to Timetabling of Elementary School Classes: A Coarse Grained Approach. *Proceedings of 22th International Symposium on Information, Communication and Automation Technologies (ICAT'09), Sarajevo, Bosnia – Herzegovina,* 1 – 5. Retrieved August 1, 2011 from IEEE Database.

Tsai, C.-C., Li, S.H.A. (2009). A Two-Stage Modeling with Genetic Algorithms fort he Nurse Scheduling Problem. *International Journal of Expert Systems with Applications 36*, 9506 – 9512. Retrieved July 6, 2009 from ScienceDirect Database.

Tseng, L.-Y., Lin, Y.-T. (2010). A Hybrid Genetic Algorithm for No – Wait Flowshop Scheduling Problem. *International Journal of Production Economics, Elsevier, Vol. 128,* 144 – 152. Retrieved August 5, 2011 from Elsevier Database.

Tseng, M.-H., Chen, S.-J., Hwang, G.-H., Shen, M.-Y. (2008). A Genetic Algorithm Rule-Based Approach for Land-Cover Classification. *Journal of Photogrammetry & Remote Sensing (ISPRS), No. 63,* 202-212. Retrieved July 1, 2008 from Elsevier Database.

Tutkun, N. (2009). Optimization of Multimodal Continuous Functions Using a New Crossover for the Real-Coded Genetic Algorithms. *International Journal of Expert Systems with Applications 36*, 8172 – 8177. Retrieved July 6, 2009 from ScienceDirect Database.

Valdez, F., Melin, P. & Castillo, O. (2011). An Improved Evolutionary Method with Fuzzy Logic for Combining Particle Swarm Optimization and Genetic Algorithms. *Applied Soft Computing 11, Elsevier,* 2625 – 2632. Retrieved April 5, 2011 from Elsevier Database.

Vazquez-Rodriguez, J.A., Petrovic, S. (2010). A New Dispatching Rule Based Genetic Algorithm for the Multi-objective Job Shop Problem. *Journal of Heuristics (16)*, 771 – 793. Springer – Verlag.

Wang, L., Tang, D.-B. (2011). An Improved Adaptive Genetic Algorithm Based on Hormone Modulation Mechanism for Job – Shop Scheduling Problem. *Expert Systems with Applications, Elsevier, Vol. 38,* 7243 – 7250. Retrieved August 5, 2011 from Elsevier Database.

Wang, L.-J., Hu, D.-W., Gong, R.-Z. (2009). Improved Genetic Algorithm for Aircraft Departure Sequencing Problem. *Proceedings of the 3th IEEE International Conference on Genetic and Evolutionary Computing,* 35 – 38. Retrieved August 5, 2011 from IEEE Database.

Wang, S. (2009). Solving Aircraft Sequencing Problem Based on Bee Evolutionary Algorithm and Clustering Method. *Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing,* 157 – 161. Retrieved August 5, 2011 from IEEE Database.

Wang, Z., Liu, J. – I. & Yu, X. (2009). Self – Fertilization Based Genetic Algorithm for University Timetabling Problem. *World Summit on Genetic and Evolutionary Computation (GEC'09),* 1001 – 1004. Shanghai, China. Retrieved August 5, 2011 from ACM Digital Library Database.

Weise, T., Zapf, M., Geihs, K. (2007). Rule-Based Genetic Programming. *Proceedings of Bio-Inspired Models of Network, Information and Computing Systems (Bionetics'07),* 8 – 15. IEEE. Retrieved January 2, 2012 from IEEExplore Database.

Wessa, P. (2012), Free Statistics Software, *Office for Research Development and Education, version 1.1.23-r7,* URL http://www.wessa.net/.

Whitley, D. (1995). Genetic Algorithms and Neural Networks. In J. Periaux, G. Winter, (Eds.). *Genetic Algorithms in Engineering and Computer Science* (203-216). Wiley & Sons Ltd.

Wong, W.K., Zeng, X.H., Au, W.M.R. (2009). A Decision Support Tool for Apparel Coordination Through Integrating the Knowledge-Based Attribute Evaluation Expert System and the T-S Fuzzy Neural Network. *Expert Systems with Applications 36, Elsevier,* 2377-2390. Retrieved June 23, 2009 from ScienceDirect Database.

Wu, J.-Z., Hao, X.-C., Chien, C.-F., Gen, M. (2011). A Novel Bi – Vector Encoding Genetic Algorithm for the Simultaneous Multiple Resources Scheduling Problem. *Journal of Intelligent Manufactoring, Springer, DOI: 10.1007/s10845-011-0570-0.* Retrieved August 5, 2011 from Springer Database.

Wu, M.-C., Shih, C.-F., Chen, C.-F. (2009). An Efficient Approach to Cross-Tab Route Planning for Water Manufacturing. *International Journal of Expert Systems with Applications 36*, 11962 – 11968. Retrieved July 6, 2009 from ScienceDirect Database.

Xiao-Feng, Z., Si-Xun, H., Zheng, S. (2010). Ray Tracing / Correlation Approach to Estimation of Surface – Based Duct Parameters from Radar Clutter. *Chinese Physics B, Vol. 19,* 1 – 8. Chinese Physical Society and IOP Publishing Ltd.

Yağmahan, B., Yenisey, M.M. (2010). A multi-objective Ant Colony System Algorithm for Flow Shop Scheduling Problem. *Expert Systems with Applications, Vol. 37*, 1361 – 1368. Retrieved February 26, 2010 from ScienceDirect Database.

Yu, S., Cao, X., Hu, M., Du, W., Zhang, J. (2009). A Real Time Schedule Method for Aircraft Landing Scheduling Problem Based on Cellular Automaton. *World Summit on Genetic and Evolutionary Computation (GEC'09),* 717 – 723. Shanghai, China. Retrieved August 5, 2011 from ACM Digital Library Database.

Yu, L., Zhao, Y., Ni, R., Zhu, Z. (2009). PM1 Steganography in JPEG Images Using Genetic Algorithm. *Journal of Soft Computing 13*, 393 – 400. Retrieved July 16, 2009 from Web of Science Database.

Yun, Y., Gen, M., Moon, C. (2010). Hybrid Genetic Algorithm with Adaptive Local Search for Precedence – Constrained Sequencing Problems. *Proceedings of the 40th IEEE Computers and Industrial Engineering (CIE'10),* 1 – 6. Retrieved August 5, 2011 from IEEE Database.

Yun, S., Lee, J., Chung, W., Kim, E., Kim, S. (2009). A Soft Computing Approach to Localization in Wireless Sensor Networks. *International Journal of Expert Systems with Applications 36*, 7552 – 7561. Retrieved July 6, 2009 from ScienceDirect Database.

Zadeh, L. A. (March 1994). Fuzzy Logic, Neural Networks, and Soft Computing. *Communications of the ACM, 37 No. 3*, 77-84.

Zhan, Z.-Z., Zhang, J., Gong, Y.-J. (2009). Ant Colony System Based on Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling. *Proceedings of 11th Annual Conference on Genetic and Evolutionary Computation Conference GECCO'09*, 1765 – 1766. Montreal, Quebec, Canada. Retrieved August 5, 2011 from ACM Digital Library Database.

Zhang, C., Tu, H. (2010). The Application of Rule Based Genetic Algorithm in Job-Shop Production Scheduling. *Proceedings of International Conference on Computing, Control and Industrial Engineering, CCIE'10*, 20 – 23. IEEE. Retrieved January 2, 2012 from IEEExplore Database.

Zheng, Y., Liu, J.-F., Geng, W.-H., Yang, J.-Y. (2009). Quantum-Inspired Genetic Evolutionary Algorithm for Course Timetabling. *Proceedings of 3rd International Conference on Genetic and Evolutionary Computing*, 750 – 753. IEEE.

APPENDICES

A. MODULE NAMES AND PREREQUISITE NUMBERS

A.1 Module Names and Prerequisite Numbers of the Training Data

Module Name	Prerequisites
1 – Giriş	-
2 – Cari Modül Giriş	1
3 – Stok Modülü Giriş	1 – 2
4 – Fatura Modülü Giriş	3
5 – Cari Modul İleri	2 – 3
6 – Stok Modul İleri	1 – 3 – 4
7 – Fatura Modul İleri 1	1 – 3 – 4 – 6
8 – Talep / Teklif Modulü	1 – 2 – 3 – 4
9 – Banka Modülü	1 – 2 – 4 – 5 – 7
10 – Müşteri Çekleri	1 – 2 – 4 – 5 – 7 – 9
11 – Borç Çekleri	1 – 2 – 4 – 8 – 9 – 10
12 – Dekont Modülü	2 – 4 – 7 – 9 – 10
13 – Entegrasyon Modülü	1 – 2 – 3 – 4 – 8 – 10 – 12
14 – Muhasebe Giriş	2 – 4 – 5 – 7 – 9 – 10 – 11
15 – Muhasebe Modülü	1 – 3 – 4 – 6 – 7 – 8 – 9 – 13 – 14
16 – Rapor Modülü	1 – 3 – 4 – 8 – 12 – 13 – 14
17 – E - Netsis	1 – 16
18 – Dinamik Depo	1 – 3 – 4 – 6 – 7 – 12 – 13 – 16
19 – Fatura Modulu İleri 2	3 – 4 – 6 – 7 – 10 – 12 – 13 – 14
20 – Dekont Özel Uygulamalar	1 – 4 – 10 – 12 – 14 – 15 – 16
21 – Dizayn Modülü	1 – 3 – 6 – 9 – 10 – 11 – 12 – 13 – 16 – 17
22 – CRM	1 – 3 – 6 – 8 – 13 – 16 – 17 – 20
23 – Netpos Modülü	1 – 2 – 5 – 8 – 10 – 11 – 12 – 14 – 16 – 17 – 21
24 – Personel Bordro Modülü	1 – 4 – 7 – 8 – 10 – 11 – 12 – 13 – 16 – 18 – 20 – 23
25 – Personel Özel Uygulamaları	1 – 24
26 – Demirbaş	1 – 3 – 6 – 13 – 18
27 – Üretim Modülü	4 – 7 – 9 – 13 – 14 – 16 – 18 – 21 – 22
28 – MRP Modulü	1 – 13 – 16 – 17 – 18 – 21 – 22 – 26
29 – İÜP / AÜP	1 – 17 – 27
30 – Maliyet Muhasebesi	1 – 14 – 15 – 17 – 25 – 29
31 – Teknik 1	17 – 21 – 23 – 28
32 – Teknik 2	17 – 31
33 – Yazılım	17 – 31 – 32
34 – Netsis Süreçleri	1 – 13 – 17 – 21 – 22 – 23 – 29
35 – Kalite Kontrol	8 – 13 – 16 – 18 – 22 – 23 – 24
36 – Kullanıcı İşlemleri	1 – 13 – 14 – 16 – 25 – 26 – 34
37 – İnsan Kaynakları	1 – 10 – 16 – 17 – 24 – 25 – 35
38 – NDI	1 – 17 – 23
39 – Dinamik Kodlama	17 – 23 – 31

A.2 Module Names and Prerequisite Numbers of the Control Data

Module Name	Prerequisites
1 –Introduction to DBMS	-
2 – DBMS-DBA	1
3 – Data Models	1
4 – Table	1 – 2
5 – Referential Integrity	1 – 2 – 4
6 – Algebraic Operators	1 – 2 – 4
7 – PK	1 – 3 – 4 – 5
8 – FK	1 – 3 – 4 – 5 – 7
9 – Normalization	1 – 4 – 5 – 7 – 8
10 – DB Design	1 – 4 – 5 – 7 – 9
11 – Data Types	3 – 4 – 10
12 – SQL-Select	3 – 4 – 5 – 7 – 8 – 11
13 – SQL-Insert	3 – 4 – 5 – 7 – 8 – 11 – 12
14 – SQL Operators	4 – 12 – 13
15 – SQL-Update	3 – 4 – 5 – 7 – 8 – 11 – 12 – 14
16 – SQL-Delete	3 – 4 – 5 – 7 – 8 – 11 – 12 – 14
17 – Sequences	4 – 7 – 8
18 – Index	4 – 7 – 8 – 11
19 – Equi-join	4 – 7 – 8 – 11 – 12 – 14
20 – Nonequi-join	4 – 7 – 8 – 11 – 12 – 14 – 19
21 – Self-join	4 – 7 – 8 – 11 – 12 – 14 – 19
22 – Subquery	4 – 7 – 8 – 12 – 14
23 – Create Table/View	3 – 4 – 5 – 7 – 8 – 11
24 – View	4 – 7 – 8 – 12 – 23
25 – Alter Table/View	3 – 4 – 5 – 7 – 8 – 11 – 23
26 – Defaults	4 – 11 – 23 – 25
27 – Rules	4 – 11 – 23 – 25
28 – Synonym	4 – 11 – 23 – 25
29 – Drop Table/View	3 – 4 – 5 – 7 – 8 – 16 – 23 – 25
30 – Trigger	4 – 5 – 7 – 8 – 11 – 12 – 13 – 15 – 16
31 – Prog. for Stored Proc.	4 – 11
32 – Stored Procedure	4 – 5 – 7 – 8 – 12 – 13 – 15 – 16 – 31
33 – Create User/Role	1 – 2
34 – Privileges	4 – 12 – 13 – 15 – 16 – 23 – 25 – 29
35 – Grant/Revoke	4 – 12 – 13 – 15 – 16 – 23 – 25 – 33 – 34
36 – System Catalog	4 – 11 – 12
37 – Transactions	12 – 13 – 15 – 16
38 – Commit/Rollback/Savepoint	12 – 13 – 15 – 16 – 37
39 – ADODB	4 – 5 – 7 – 8 – 11 – 12 – 13 – 15 – 16 – 23 – 25 – 29
40 – SQL Server	4 – 5 – 7 – 8 – 11 – 12 – 13 – 15 – 16 – 23 – 25 – 29
41 – MS Access	4 – 5 – 7 – 8 – 11 – 12 – 13 – 15 – 16 – 23 – 25 – 29
42 – XML with DB	4 – 12 – 13

B. LOGICAL REPRESENTATION OF RULES

B.1 Rules for The Training Data

- **IF** M1 **THEN** M2
- **IF** M1 **AND** M2 **THEN** M3
- **IF** M3 **THEN** M4
- **IF** M2 **AND** M3 **THEN** M5
- **IF** M1 **AND** M3 **AND** M4 **THEN** M6
- **IF** M1 **AND** M3 **AND** M4 **AND** M6 **THEN** M7
- **IF** M1 **AND** M2 **AND** M3 **AND** M4 **THEN** M8
- **IF** M1 **AND** M2 **AND** M4 **AND** M5 **AND** M7 **THEN** M9
- **IF** M1 **AND** M2 **AND** M4 **AND** M5 **AND** M7 **AND** M9 **THEN** M10
- **IF** M1 **AND** M2 **AND** M4 **AND** M8 **AND** M9 **AND** M10 **THEN** M11
- **IF** M2 **AND** M4 **AND** M7 **AND** M9 **AND** M10 **THEN** M12
- **IF** M1 **AND** M2 **AND** M3 **AND** M4 **AND** M8 **AND** M10 **AND** M12 **THEN** M13
- **IF** M2 **AND** M4 **AND** M5 **AND** M7 **AND** M9 **AND** M10 **AND** M11 **THEN** M14
- **IF** M1 **AND** M3 **AND** M4 **AND** M6 **AND** M7 **AND** M8 **AND** M9 **AND** M13 **AND** M14 **THEN** M15
- **IF** M1 M3 **AND** M4 **AND** M8 **AND** M12 **AND** M13 **AND** M14 **THEN** M16
- **IF** M1 **AND** M16 **THEN** M17
- **IF** M1 **AND** M3 **AND** M4 **AND** M6 **AND** M7 **AND** M12 **AND** M13 **AND** M16 **THEN** M18
- **IF** M3 **AND** M4 **AND** M6 **AND** M7 **AND** M10 **AND** M12 **AND** M13 **AND** M14 **THEN** M19
- **IF** M1 **AND** M4 **AND** M10 **AND** M12 **AND** M14 **AND** M15 **AND** M16 **THEN** M20
- **IF** M1 **AND** M3 **AND** M6 **AND** M9 **AND** M10 **AND** M11 **AND** M12 **AND** M13 **AND** M16 **AND** M17 **THEN** M21
- **IF** M1 **AND** M3 **AND** M6 **AND** M8 **AND** M13 **AND** M16 **AND** M17 **AND** M20 **THEN** M22
- **IF** M1 **AND** M2 **AND** M5 **AND** M8 **AND** M10 **AND** M11 **AND** M12 **AND** M14 **AND** M16 **AND** M17 **AND** M21 **THEN** M23
- **IF** M1 **AND** M4 **AND** M7 **AND** M8 **AND** M10 **AND** M11 **AND** M12 **AND** M13 **AND** M16 **AND** M18 **AND** M20 **AND** M23 **THEN** M24
- **IF** M1 **AND** M24 **THEN** M25
- **IF** M1 **AND** M3 **AND** M6 **AND** M13 **AND** M18 **THEN** M26
- **IF** M4 **AND** M7 **AND** M9 **AND** M13 **AND** M14 **AND** M16 **AND** M18 **AND** M21 **AND** M22 **THEN** M27
- **IF** M1 **AND** M13 **AND** M16 **AND** M17 **AND** M18 **AND** M21 **AND** M22 **AND** M27 **THEN** M28
- **IF** M1 **AND** M17 **AND** M27 **THEN** M29
- **IF** M1 **AND** M14 **AND** M15 **AND** M17 **AND** M25 **AND** M29 **THEN** M30
- **IF** M17 **AND** M21 **AND** M23 **AND** M28 **THEN** M31
- **IF** M17 **AND** M31 **THEN** M32
- **IF** M17 **AND** M31 **AND** M32 **THEN** M33
- **IF** M1 **AND** M13 **AND** M17 **AND** M21 **AND** M22 **AND** M23 **AND** M29 **THEN** M34
- **IF** M8 **AND** M13 **AND** M16 **AND** M18 **AND** M22 **AND** M23 **AND** M24 **THEN** M35
- **IF** M1 **AND** M13 **AND** M14 **AND** M16 **AND** M25 **AND** M26 **AND** M34 **THEN** M36
- **IF** M1 **AND** M10 **AND** M16 **AND** M17 **AND** M24 **AND** M25 **AND** M35 **THEN** M37
- **IF** M1 **AND** M17 **AND** M23 **THEN** M38
- **IF** M17 **AND** M23 **AND** M31 **AND** M32 **AND** M33 **THEN** M39

B.2 Rules for The Control Data

- **IF** M1 **THEN** M2
- **IF** M1 **THEN** M3
- **IF** M1 **AND** M2 **THEN** M4
- **IF** M1 **AND** M2 **AND** M4 **THEN** M5
- **IF** M1 **AND** M2 **AND** M4 **THEN** M6
- **IF** M1 **AND** M3 **AND** M4 **AND** M5 **THEN** M7
- **IF** M1 **AND** M3 **AND** M5 **AND** M5 AND M7 **THEN** M8
- **IF** M1 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **THEN** M9
- **IF** M1 **AND** M3 **AND** M4 **AND** M5 **AND** M9 **THEN** M10
- **IF** M3 **AND** M4 **AND** M10 **THEN** M11
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M11 **THEN** M12
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M11 AND M12 **THEN** M13
- **IF** M4 **AND** M12 **AND** M13 **THEN** M14
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M14 **THEN** M15
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M14 **THEN** M16
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **THEN** M17
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **THEN** M18
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M14 **THEN** M19
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M14 **AND** M19 **THEN** M20
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M14 **AND** M19 **THEN** M21
- **IF** M4 **AND** M7 **AND** M8 **AND** M12 **AND** M14 **THEN** M22
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M11 **THEN** M23
- **IF** M4 **AND** M7 **AND** M8 **AND** M12 **AND** M23 **THEN** M24
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 AND M8 AND M11 **AND** M23 **THEN** M25
- **IF** M4 **AND** M11 **AND** M23 **AND** M25 **THEN** M26
- **IF** M4 **AND** M11 **AND** M23 **AND** M25 **THEN** M27
- **IF** M4 **AND** M11 **AND** M23 **AND** M25 **THEN** M28
- **IF** M3 **AND** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M16 **AND** M23 **AND** M25 **THEN** M29
- **IF** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **THEN** M30
- **IF** M4 **AND** M11 **THEN** M31
- **IF** M4 **AND** M5 **AND** M7 **AND** M8 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **AND** M31 **THEN** M32
- **IF** M1 **AND** M2 **THEN** M33
- **IF** M4 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **AND** M23 **AND** M25 **AND** M29 **THEN** M34
- **IF** M4 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **AND** M23 **AND** M25 **AND** M29 **AND** M33 **AND** M34 **THEN** M35
- **IF** M4 **AND** M11 **AND** M12 **THEN** M36
- **IF** M12 **AND** M13 **AND** M15 **AND** M16 **THEN** M37
- **IF** M12 **AND** M13 **AND** M15 **AND** M16 AND M37 **THEN** M38
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **AND** M23 **AND** M25 **AND** M29 **THEN** M39
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **AND** M23 **AND** M25 **AND** M29 **THEN** M40
- **IF** M4 **AND** M7 **AND** M8 **AND** M11 **AND** M12 **AND** M13 **AND** M15 **AND** M16 **AND** M23 **AND** M25 **AND** M29 **THEN** M41
- **IF** M4 **AND** M12 **AND** M13 **THEN** M42

C. XML Files of the Rules for the Training and Control Data

C.1 XML of Training Data

```
<?xml version="1.0"?>
<modules>
  <module mno="0" hour="4" />
  <module mno="1" hour="6">
    <prereq>0</prereq>
  </module>
  <module mno="2" hour="12">
    <prereq>0</prereq>
    <prereq>1</prereq>
  </module>
  <module mno="3" hour="3">
    <prereq>2</prereq>
  </module>
  <module mno="4" hour="2">
    <prereq>1</prereq>
    <prereq>2</prereq>
  </module>
  <module mno="5" hour="16">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
  </module>
  <module mno="6" hour="8">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>5</prereq>
  </module>
  <module mno="7" hour="4">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
  </module>
  <module mno="8" hour="4">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
    <prereq>6</prereq>
  </module>
  <module mno="9" hour="6">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
    <prereq>6</prereq>
    <prereq>8</prereq>
  </module>
  <module mno="10" hour="8">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>3</prereq>
    <prereq>7</prereq>
    <prereq>8</prereq>
    <prereq>9</prereq>
  </module>
  <module mno="11" hour="3">
    <prereq>1</prereq>
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>8</prereq>
  <prereq>9</prereq>
</module>
<module mno="12" hour="12">
  <prereq>0</prereq>
  <prereq>1</prereq>
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>7</prereq>
  <prereq>9</prereq>
  <prereq>11</prereq>
</module>
<module mno="13" hour="10">
  <prereq>1</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>8</prereq>
  <prereq>9</prereq>
  <prereq>10</prereq>
</module>
<module mno="14" hour="3">
  <prereq>0</prereq>
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>5</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>8</prereq>
  <prereq>12</prereq>
  <prereq>13</prereq>
</module>
<module mno="15" hour="2">
  <prereq>0</prereq>
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>7</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>13</prereq>
</module>
<module mno="16" hour="4">
  <prereq>0</prereq>
  <prereq>15</prereq>
</module>
<module mno="17" hour="4">
  <prereq>0</prereq>
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>5</prereq>
  <prereq>6</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>15</prereq>
</module>
<module mno="18" hour="8">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>5</prereq>
  <prereq>6</prereq>
  <prereq>9</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>13</prereq>
</module>
<module mno="19" hour="6">
  <prereq>0</prereq>
  <prereq>3</prereq>
  <prereq>9</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
  <prereq>14</prereq>
  <prereq>15</prereq>
</module>
<module mno="20" hour="8">
  <prereq>0</prereq>
  <prereq>2</prereq>
  <prereq>5</prereq>
  <prereq>8</prereq>
  <prereq>9</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>15</prereq>
  <prereq>16</prereq>
</module>
<module mno="21" hour="4">
  <prereq>0</prereq>
  <prereq>2</prereq>
  <prereq>5</prereq>
  <prereq>7</prereq>
  <prereq>12</prereq>
  <prereq>15</prereq>
  <prereq>16</prereq>
  <prereq>19</prereq>
</module>
<module mno="22" hour="2">
  <prereq>0</prereq>
  <prereq>1</prereq>
  <prereq>4</prereq>
  <prereq>7</prereq>
  <prereq>9</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
  <prereq>15</prereq>
  <prereq>16</prereq>
  <prereq>20</prereq>
</module>
<module mno="23" hour="12">
  <prereq>0</prereq>
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>9</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>15</prereq>
  <prereq>17</prereq>
  <prereq>19</prereq>
  <prereq>22</prereq>
</module>
<module mno="24" hour="16">
  <prereq>0</prereq>
  <prereq>23</prereq>
</module>
<module mno="25" hour="3">
```

```
  <prereq>0</prereq>
  <prereq>2</prereq>
  <prereq>5</prereq>
  <prereq>12</prereq>
  <prereq>17</prereq>
</module>
<module mno="26" hour="4">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>8</prereq>
  <prereq>12</prereq>
  <prereq>13</prereq>
  <prereq>15</prereq>
  <prereq>17</prereq>
  <prereq>20</prereq>
  <prereq>21</prereq>
</module>
<module mno="27" hour="6">
  <prereq>0</prereq>
  <prereq>12</prereq>
  <prereq>15</prereq>
  <prereq>16</prereq>
  <prereq>17</prereq>
  <prereq>20</prereq>
  <prereq>21</prereq>
  <prereq>25</prereq>
</module>
<module mno="28" hour="2">
  <prereq>0</prereq>
  <prereq>16</prereq>
  <prereq>26</prereq>
</module>
<module mno="29" hour="4">
  <prereq>0</prereq>
  <prereq>13</prereq>
  <prereq>14</prereq>
  <prereq>16</prereq>
  <prereq>24</prereq>
  <prereq>28</prereq>
</module>
<module mno="30" hour="4">
  <prereq>16</prereq>
  <prereq>20</prereq>
  <prereq>22</prereq>
  <prereq>27</prereq>
</module>
<module mno="31" hour="3">
  <prereq>16</prereq>
  <prereq>30</prereq>
</module>
<module mno="32" hour="4">
  <prereq>16</prereq>
  <prereq>30</prereq>
  <prereq>31</prereq>
</module>
<module mno="33" hour="2">
  <prereq>0</prereq>
  <prereq>12</prereq>
  <prereq>16</prereq>
  <prereq>20</prereq>
  <prereq>21</prereq>
  <prereq>22</prereq>
  <prereq>28</prereq>
</module>
<module mno="34" hour="4">
```

```
    <prereq>7</prereq>
    <prereq>12</prereq>
    <prereq>15</prereq>
    <prereq>17</prereq>
    <prereq>21</prereq>
    <prereq>22</prereq>
    <prereq>23</prereq>
  </module>
  <module mno="35" hour="2">
    <prereq>0</prereq>
    <prereq>12</prereq>
    <prereq>13</prereq>
    <prereq>15</prereq>
    <prereq>24</prereq>
    <prereq>25</prereq>
    <prereq>33</prereq>
  </module>
  <module mno="36" hour="4">
    <prereq>0</prereq>
    <prereq>9</prereq>
    <prereq>15</prereq>
    <prereq>16</prereq>
    <prereq>23</prereq>
    <prereq>24</prereq>
    <prereq>34</prereq>
  </module>
  <module mno="37" hour="3">
    <prereq>0</prereq>
    <prereq>16</prereq>
    <prereq>22</prereq>
  </module>
  <module mno="38" hour="4">
    <prereq>16</prereq>
    <prereq>22</prereq>
    <prereq>30</prereq>
    <prereq>31</prereq>
    <prereq>32</prereq>
  </module>
</modules>
```

C.2 XML of Control Data

```
<?xml version="1.0"?>
<modules>
  <module mno="0" hour="2" />
  <module mno="1" hour="2">
    <prereq>0</prereq>
  </module>
  <module mno="2" hour="3">
    <prereq>0</prereq>
  </module>
  <module mno="3" hour="4">
    <prereq>0</prereq>
    <prereq>1</prereq>
  </module>
  <module mno="4" hour="2">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>3</prereq>
  </module>
  <module mno="5" hour="4">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>3</prereq>
  </module>
  <module mno="6" hour="3">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
  </module>
  <module mno="7" hour="2">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
    <prereq>6</prereq>
 </module>
  <module mno="8" hour="4">
    <prereq>0</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
    <prereq>6</prereq>
    <prereq>7</prereq>
  </module>
  <module mno="9" hour="8">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
    <prereq>8</prereq>
  </module>
  <module mno="10" hour="4">
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>9</prereq>
  </module>
  <module mno="11" hour="4">
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>4</prereq>
    <prereq>6</prereq>
    <prereq>7</prereq>
    <prereq>10</prereq>
```

```
</module>
<module mno="12" hour="2">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
</module>
<module mno="13" hour="4">
  <prereq>3</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
</module>
<module mno="14" hour="2">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
</module>
<module mno="15" hour="2">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
</module>
<module mno="16" hour="1">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
</module>
<module mno="17" hour="1">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
</module>
<module mno="18" hour="3">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
</module>
<module mno="19" hour="3">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
  <prereq>18</prereq>
</module>
<module mno="20" hour="2">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
  <prereq>18</prereq>
</module>
<module mno="21" hour="6">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>11</prereq>
  <prereq>13</prereq>
</module>
<module mno="22" hour="3">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
</module>
<module mno="23" hour="2">
  <prereq>3</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>11</prereq>
  <prereq>22</prereq>
</module>
<module mno="24" hour="2">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>22</prereq>
</module>
<module mno="25" hour="1">
  <prereq>3</prereq>
  <prereq>10</prereq>
  <prereq>22</prereq>
  <prereq>24</prereq>
</module>
<module mno="26" hour="1">
  <prereq>3</prereq>
  <prereq>10</prereq>
  <prereq>22</prereq>
  <prereq>24</prereq>
</module>
<module mno="27" hour="1">
  <prereq>3</prereq>
  <prereq>10</prereq>
  <prereq>22</prereq>
  <prereq>24</prereq>
</module>
<module mno="28" hour="1">
  <prereq>2</prereq>
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>15</prereq>
  <prereq>22</prereq>
  <prereq>24</prereq>
```

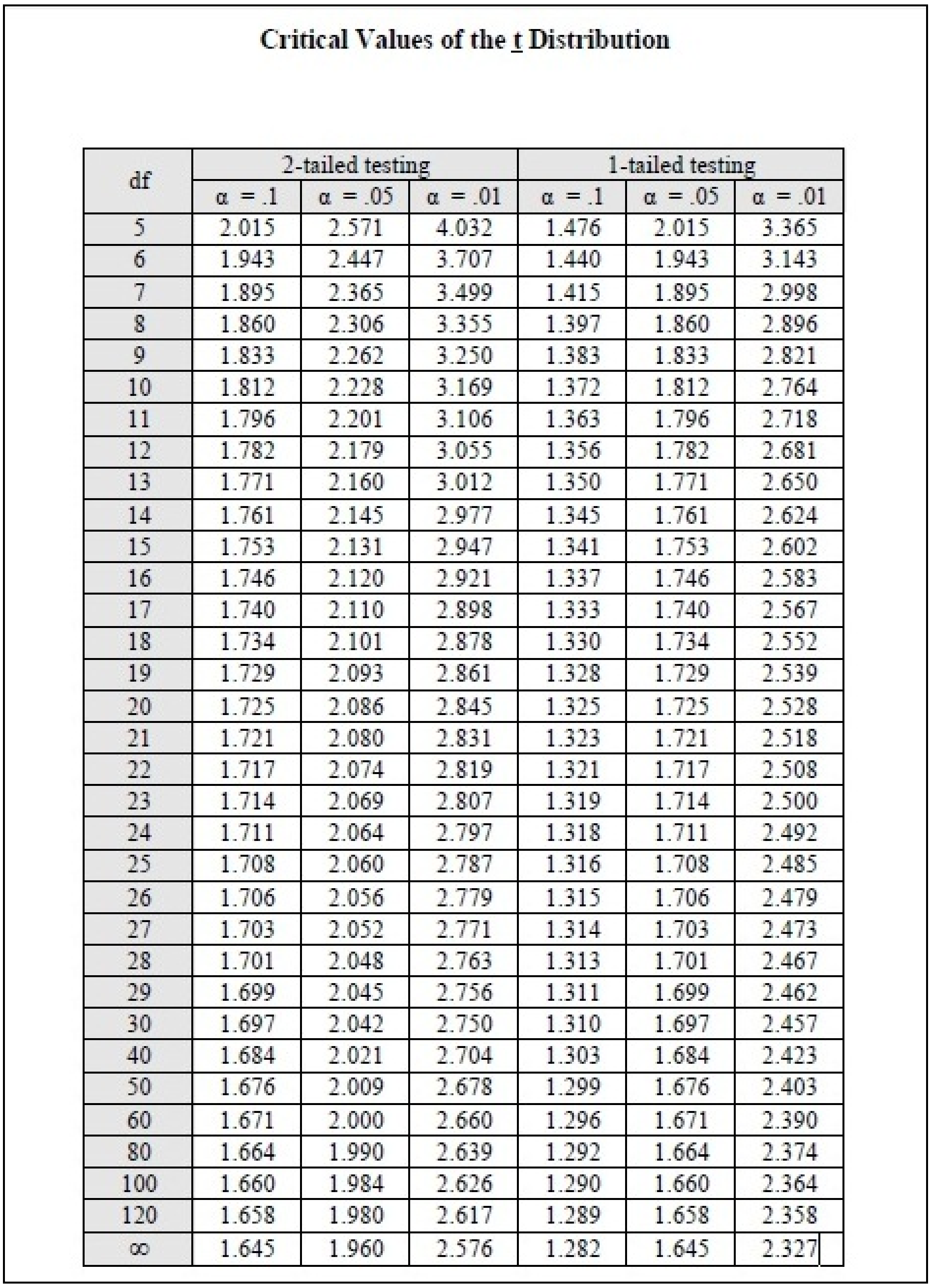
```
</module>
<module mno="29" hour="2">
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>14</prereq>
  <prereq>15</prereq>
</module>
<module mno="30" hour="2">
  <prereq>3</prereq>
  <prereq>10</prereq>
</module>
<module mno="31" hour="3">
  <prereq>3</prereq>
  <prereq>4</prereq>
  <prereq>6</prereq>
  <prereq>7</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>14</prereq>
  <prereq>15</prereq>
  <prereq>30</prereq>
</module>
<module mno="32" hour="3">
  <prereq>0</prereq>
  <prereq>1</prereq>
</module>
<module mno="33" hour="2">
  <prereq>3</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>14</prereq>
  <prereq>15</prereq>
  <prereq>22</prereq>
  <prereq>24</prereq>
  <prereq>28</prereq>
</module>
<module mno="34" hour="2">
  <prereq>3</prereq>
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>14</prereq>
  <prereq>15</prereq>
  <prereq>22</prereq>
  <prereq>24</prereq>
  <prereq>28</prereq>
  <prereq>32</prereq>
  <prereq>33</prereq>
</module>
<module mno="35" hour="3">
  <prereq>3</prereq>
  <prereq>10</prereq>
  <prereq>11</prereq>
</module>
<module mno="36" hour="2">
  <prereq>11</prereq>
  <prereq>12</prereq>
  <prereq>14</prereq>
  <prereq>15</prereq>
</module>
<module mno="37" hour="2">
```

```xml
    <prereq>11</prereq>
    <prereq>12</prereq>
    <prereq>14</prereq>
    <prereq>15</prereq>
    <prereq>36</prereq>
  </module>
  <module mno="38" hour="4">
    <prereq>3</prereq>
    <prereq>6</prereq>
    <prereq>7</prereq>
    <prereq>10</prereq>
    <prereq>11</prereq>
    <prereq>12</prereq>
    <prereq>14</prereq>
    <prereq>15</prereq>
    <prereq>22</prereq>
    <prereq>24</prereq>
    <prereq>28</prereq>
  </module>
  <module mno="39" hour="4">
    <prereq>3</prereq>
    <prereq>6</prereq>
    <prereq>7</prereq>
    <prereq>10</prereq>
    <prereq>11</prereq>
    <prereq>12</prereq>
    <prereq>14</prereq>
    <prereq>15</prereq>
    <prereq>22</prereq>
    <prereq>24</prereq>
    <prereq>28</prereq>
  </module>
  <module mno="40" hour="6">
    <prereq>3</prereq>
    <prereq>6</prereq>
    <prereq>7</prereq>
    <prereq>10</prereq>
    <prereq>11</prereq>
    <prereq>12</prereq>
    <prereq>14</prereq>
    <prereq>15</prereq>
    <prereq>22</prereq>
    <prereq>24</prereq>
    <prereq>28</prereq>
  </module>
  <module mno="41" hour="6">
    <prereq>3</prereq>
    <prereq>11</prereq>
    <prereq>12</prereq>
  </module>
</modules>
```

D. Sample XML File of the Program

```
<?xml version="1.0"?>
<program>
        <day dno="0">
                <ders saat="1">FATURA MODÜLÜ GIRIS</ders>
                <ders saat="2">FATURA MODÜLÜ GIRIS</ders>
                <ders saat="3">FATURA MODÜLÜ GIRIS</ders>
                <ders saat="4" />
                <ders saat="5">GIRIS</ders>
                <ders saat="6">GIRIS</ders>
                <ders saat="7">GIRIS</ders>
                <ders saat="8">GIRIS</ders>
        </day>
        <day dno="1">
                <ders saat="1">BANKA MODÜLÜ</ders>
                <ders saat="2">BANKA MODÜLÜ</ders>
                <ders saat="3">BANKA MODÜLÜ</ders>
                <ders saat="4">BANKA MODÜLÜ</ders>
                <ders saat="5">TALEP / TEKLIF  MODULÜ</ders>
                <ders saat="6">TALEP / TEKLIF  MODULÜ</ders>
                <ders saat="7">TALEP / TEKLIF  MODULÜ</ders>
                <ders saat="8">TALEP / TEKLIF  MODULÜ</ders>
        </day>
        <day dno="2">
                <ders saat="1">MÜSTERI ÇEKLERI</ders>
                <ders saat="2">MÜSTERI ÇEKLERI</ders>
                <ders saat="3">MÜSTERI ÇEKLERI</ders>
                <ders saat="4">MÜSTERI ÇEKLERI</ders>
                <ders saat="5">MÜSTERI ÇEKLERI</ders>
                <ders saat="6">MÜSTERI ÇEKLERI</ders>
                <ders saat="7">STOK MODUL ILERI</ders>
                <ders saat="8">STOK MODUL ILERI</ders>
        </day>
        <day dno="3">
                <ders saat="1">STOK MODUL ILERI</ders>
                <ders saat="2">STOK MODUL ILERI</ders>
                <ders saat="3">STOK MODUL ILERI</ders>
                <ders saat="4">STOK MODUL ILERI</ders>
                <ders saat="5">STOK MODUL ILERI</ders>
                <ders saat="6">STOK MODUL ILERI</ders>
                <ders saat="7">STOK MODUL ILERI</ders>
                <ders saat="8">STOK MODUL ILERI</ders>
        </day>
        <day dno="4">
                <ders saat="1">STOK MODUL ILERI</ders>
                <ders saat="2">STOK MODUL ILERI</ders>
                <ders saat="3">STOK MODUL ILERI</ders>
                <ders saat="4">STOK MODUL ILERI</ders>
                <ders saat="5">STOK MODUL ILERI</ders>
                <ders saat="6">STOK MODUL ILERI</ders>
                <ders saat="7">MUHASEBE GIRIS  </ders>
                <ders saat="8">MUHASEBE GIRIS  </ders>
        </day>
        <day dno="5">
                <ders saat="1">MUHASEBE GIRIS  </ders>
                <ders saat="2">MUHASEBE GIRIS  </ders>
                <ders saat="3">MUHASEBE GIRIS  </ders>
                <ders saat="4">MUHASEBE GIRIS  </ders>
                <ders saat="5">MUHASEBE GIRIS  </ders>
                <ders saat="6">MUHASEBE GIRIS  </ders>
                <ders saat="7">MUHASEBE GIRIS  </ders>
                <ders saat="8">MUHASEBE GIRIS  </ders>
        </day>
        <day dno="6">
                <ders saat="1">TEKNIK  2</ders>
                <ders saat="2">TEKNIK  2</ders>
                <ders saat="3">TEKNIK  2</ders>
                <ders saat="4" />
                <ders saat="5">NETPOS MODÜLÜ</ders>
                <ders saat="6">NETPOS MODÜLÜ</ders>
                <ders saat="7">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="8">ENTEGRASYON MODÜLÜ</ders>
        </day>
        <day dno="7">
                <ders saat="1">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="2">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="3">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="4">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="5">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="6">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="7">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="8">ENTEGRASYON MODÜLÜ</ders>
        </day>
        <day dno="8">
                <ders saat="1">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="2">ENTEGRASYON MODÜLÜ</ders>
                <ders saat="3">IÜP / AÜP</ders>
                <ders saat="4">IÜP / AÜP</ders>
                <ders saat="5">DEMIRBAS</ders>
                <ders saat="6">DEMIRBAS</ders>
                <ders saat="7">DEMIRBAS</ders>
                <ders saat="8" />
        </day>
        <day dno="9">
                <ders saat="1">NDI</ders>
                <ders saat="2">NDI</ders>
                <ders saat="3">NDI</ders>
                <ders saat="4" />
                <ders saat="5">MUHASEBE MODÜLÜ</ders>
                <ders saat="6">MUHASEBE MODÜLÜ</ders>
                <ders saat="7">MUHASEBE MODÜLÜ</ders>
                <ders saat="8" />
        </day>
        <day dno="10">
                <ders saat="1">RAPOR MODÜLÜ</ders>
                <ders saat="2">RAPOR MODÜLÜ</ders>
                <ders saat="3">DEKONT ÖZEL UYGULAMALAR  </ders>
                <ders saat="4">DEKONT ÖZEL UYGULAMALAR  </ders>
                <ders saat="5">DEKONT ÖZEL UYGULAMALAR  </ders>
                <ders saat="6">DEKONT ÖZEL UYGULAMALAR  </ders>
                <ders saat="7">DEKONT ÖZEL UYGULAMALAR  </ders>
                <ders saat="8">DEKONT ÖZEL UYGULAMALAR  </ders>
        </day>
        <day dno="11">
                <ders saat="1">KALITE KONTROL</ders>
                <ders saat="2">KALITE KONTROL</ders>
                <ders saat="3">KALITE KONTROL</ders>
                <ders saat="4">KALITE KONTROL</ders>
                <ders saat="5" />
                <ders saat="6" />
                <ders saat="7" />
                <ders saat="8" />
        </day>
</program>
```

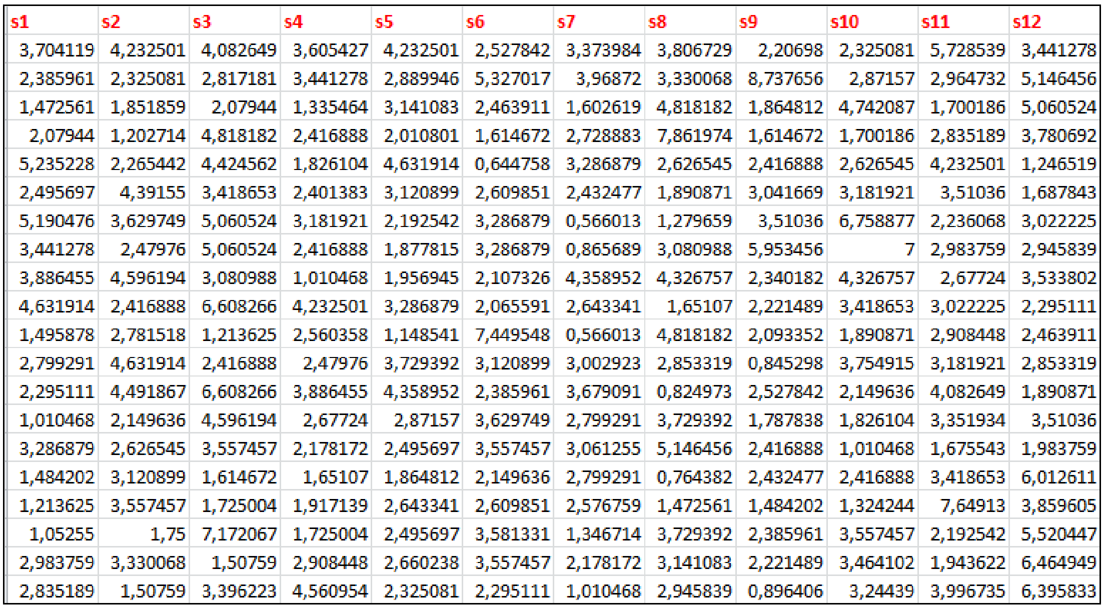
E. CRITICAL VALUES OF THE T DISTRIBUTION



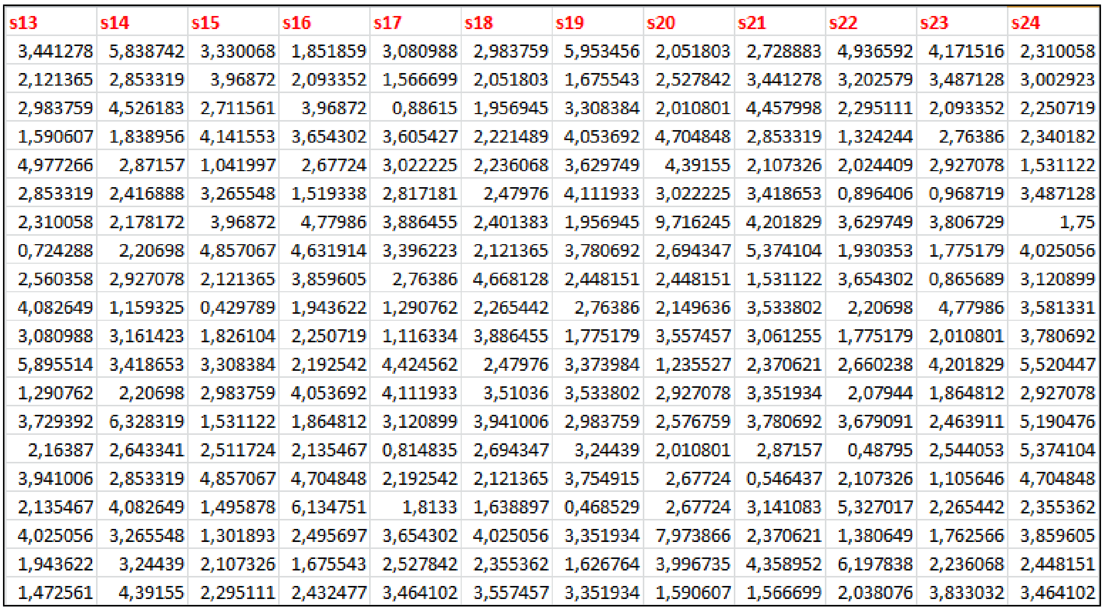
Critical Values of the t Distribution

df	2-tailed testing			1-tailed testing		
	α = .1	α = .05	α = .01	α = .1	α = .05	α = .01
5	2.015	2.571	4.032	1.476	2.015	3.365
6	1.943	2.447	3.707	1.440	1.943	3.143
7	1.895	2.365	3.499	1.415	1.895	2.998
8	1.860	2.306	3.355	1.397	1.860	2.896
9	1.833	2.262	3.250	1.383	1.833	2.821
10	1.812	2.228	3.169	1.372	1.812	2.764
11	1.796	2.201	3.106	1.363	1.796	2.718
12	1.782	2.179	3.055	1.356	1.782	2.681
13	1.771	2.160	3.012	1.350	1.771	2.650
14	1.761	2.145	2.977	1.345	1.761	2.624
15	1.753	2.131	2.947	1.341	1.753	2.602
16	1.746	2.120	2.921	1.337	1.746	2.583
17	1.740	2.110	2.898	1.333	1.740	2.567
18	1.734	2.101	2.878	1.330	1.734	2.552
19	1.729	2.093	2.861	1.328	1.729	2.539
20	1.725	2.086	2.845	1.325	1.725	2.528
21	1.721	2.080	2.831	1.323	1.721	2.518
22	1.717	2.074	2.819	1.321	1.717	2.508
23	1.714	2.069	2.807	1.319	1.714	2.500
24	1.711	2.064	2.797	1.318	1.711	2.492
25	1.708	2.060	2.787	1.316	1.708	2.485
26	1.706	2.056	2.779	1.315	1.706	2.479
27	1.703	2.052	2.771	1.314	1.703	2.473
28	1.701	2.048	2.763	1.313	1.701	2.467
29	1.699	2.045	2.756	1.311	1.699	2.462
30	1.697	2.042	2.750	1.310	1.697	2.457
40	1.684	2.021	2.704	1.303	1.684	2.423
50	1.676	2.009	2.678	1.299	1.676	2.403
60	1.671	2.000	2.660	1.296	1.671	2.390
80	1.664	1.990	2.639	1.292	1.664	2.374
100	1.660	1.984	2.626	1.290	1.660	2.364
120	1.658	1.980	2.617	1.289	1.658	2.358
∞	1.645	1.960	2.576	1.282	1.645	2.327

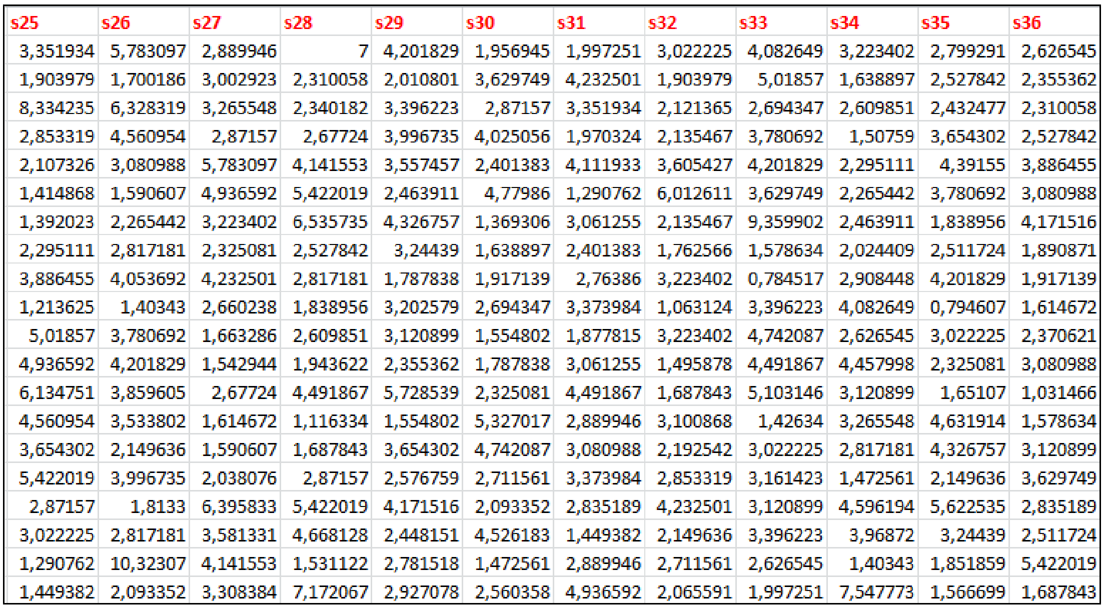
F. Best t Values of SDST for All Scenarios of 100 Individuals



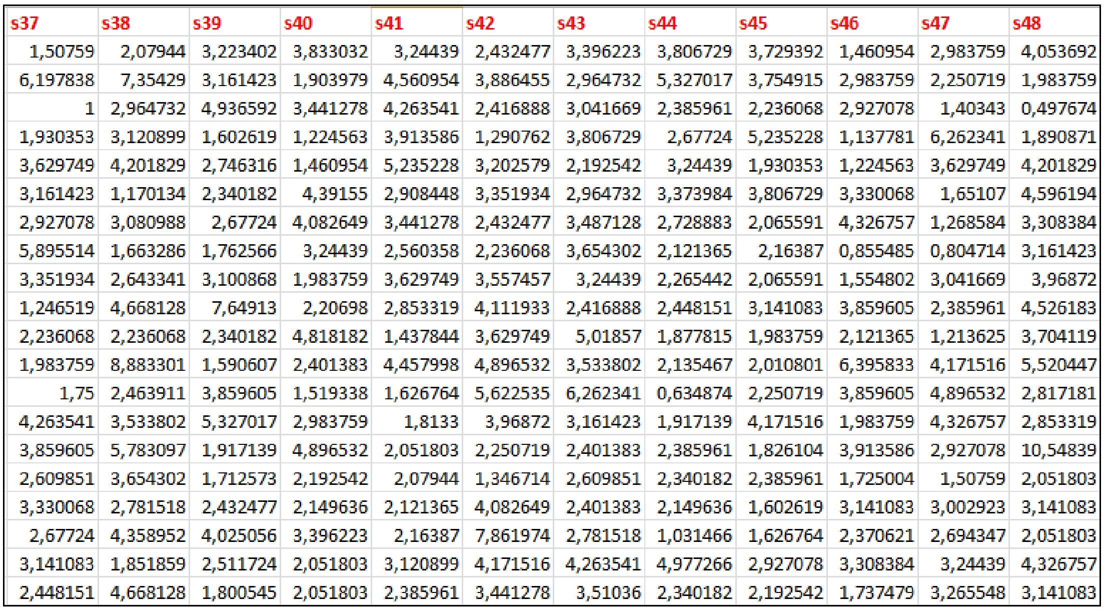
s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
3,704119	4,232501	4,082649	3,605427	4,232501	2,527842	3,373984	3,806729	2,20698	2,325081	5,728539	3,441278
2,385961	2,325081	2,817181	3,441278	2,889946	5,327017	3,96872	3,330068	8,737656	2,87157	2,964732	5,146456
1,472561	1,851859	2,07944	1,335464	3,141083	2,463911	1,602619	4,818182	1,864812	4,742087	1,700186	5,060524
2,07944	1,202714	4,818182	2,416888	2,010801	1,614672	2,728883	7,861974	1,614672	1,700186	2,835189	3,780692
5,235228	2,265442	4,424562	1,826104	4,631914	0,644758	3,286879	2,626545	2,416888	2,626545	4,232501	1,246519
2,495697	4,39155	3,418653	2,401383	3,120899	2,609851	2,432477	1,890871	3,041669	3,181921	3,51036	1,687843
5,190476	3,629749	5,060524	3,181921	2,192542	3,286879	0,566013	1,279659	3,51036	6,758877	2,236068	3,022225
3,441278	2,47976	5,060524	2,416888	1,877815	3,286879	0,865689	3,080988	5,953456	7	2,983759	2,945839
3,886455	4,596194	3,080988	1,010468	1,956945	2,107326	4,358952	4,326757	2,340182	4,326757	2,67724	3,533802
4,631914	2,416888	6,608266	4,232501	3,286879	2,065591	2,643341	1,65107	2,221489	3,418653	3,022225	2,295111
1,495878	2,781518	1,213625	2,560358	1,148541	7,449548	0,566013	4,818182	2,093352	1,890871	2,908448	2,463911
2,799291	4,631914	2,416888	2,47976	3,729392	3,120899	3,002923	2,853319	0,845298	3,754915	3,181921	2,853319
2,295111	4,491867	6,608266	3,886455	4,358952	2,385961	3,679091	0,824973	2,527842	2,149636	4,082649	1,890871
1,010468	2,149636	4,596194	2,67724	2,87157	3,629749	2,799291	3,729392	1,787838	1,826104	3,351934	3,51036
3,286879	2,626545	3,557457	2,178172	2,495697	3,557457	3,061255	5,146456	2,416888	1,010468	1,675543	1,983759
1,484202	3,120899	1,614672	1,65107	1,864812	2,149636	2,799291	0,764382	2,432477	2,416888	3,418653	6,012611
1,213625	3,557457	1,725004	1,917139	2,643341	2,609851	2,576759	1,472561	1,484202	1,324244	7,64913	3,859605
1,05255	1,75	7,172067	1,725004	2,495697	3,581331	1,346714	3,729392	2,385961	3,557457	2,192542	5,520447
2,983759	3,330068	1,50759	2,908448	2,660238	3,557457	2,178172	3,141083	2,221489	3,464102	1,943622	6,464949
2,835189	1,50759	3,396223	4,560954	2,325081	2,295111	1,010468	2,945839	0,896406	3,24439	3,996735	6,395833



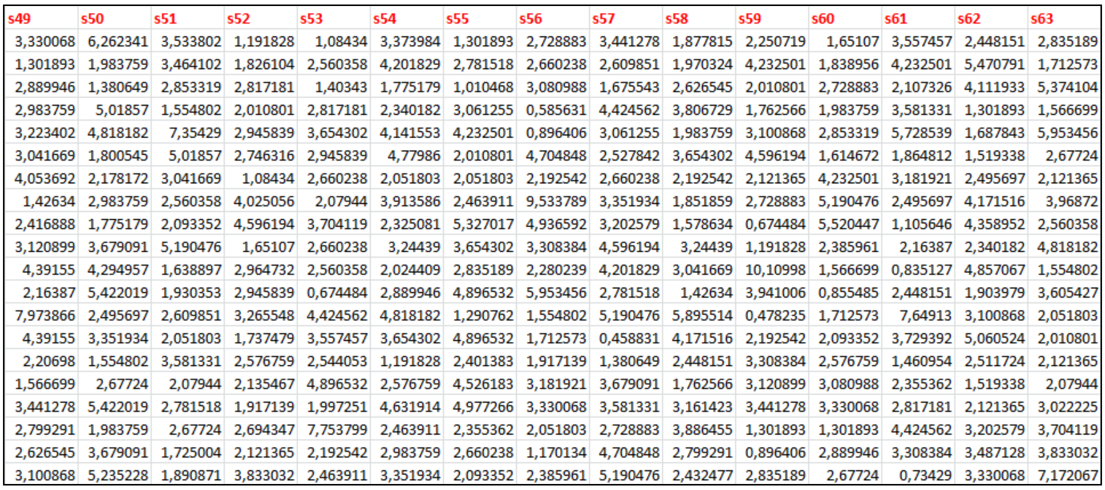
s13	s14	s15	s16	s17	s18	s19	s20	s21	s22	s23	s24
3,441278	5,838742	3,330068	1,851859	3,080988	2,983759	5,953456	2,051803	2,728883	4,936592	4,171516	2,310058
2,121365	2,853319	3,96872	2,093352	1,566699	2,051803	1,675543	2,527842	3,441278	3,202579	3,487128	3,002923
2,983759	4,526183	2,711561	3,96872	0,88615	1,956945	3,308384	2,010801	4,457998	2,295111	2,093352	2,250719
1,590607	1,838956	4,141553	3,654302	3,605427	2,221489	4,053692	4,704848	2,853319	1,324244	2,76386	2,340182
4,977266	2,87157	1,041997	2,67724	3,022225	2,236068	3,629749	4,39155	2,107326	2,024409	2,927078	1,531122
2,853319	2,416888	3,265548	1,519338	2,817181	2,47976	4,111933	3,022225	3,418653	0,896406	0,968719	3,487128
2,310058	2,178172	3,96872	4,77986	3,886455	2,401383	1,956945	9,716245	4,201829	3,629749	3,806729	1,75
0,724288	2,20698	4,857067	4,631914	3,396223	2,121365	3,780692	2,694347	5,374104	1,930353	1,775179	4,025056
2,560358	2,927078	2,121365	3,859605	2,76386	4,668128	2,448151	2,448151	1,531122	3,654302	0,865689	3,120899
4,082649	1,159325	0,429789	1,943622	1,290762	2,265442	2,76386	2,149636	3,533802	2,20698	4,77986	3,581331
3,080988	3,161423	1,826104	2,250719	1,116334	3,886455	1,775179	3,557457	3,061255	1,775179	2,010801	3,780692
5,895514	3,418653	3,308384	2,192542	4,424562	2,47976	3,373984	1,235527	2,370621	2,660238	4,201829	5,520447
1,290762	2,20698	2,983759	4,053692	4,111933	3,51036	3,533802	2,927078	3,351934	2,07944	1,864812	2,927078
3,729392	6,328319	1,531122	1,864812	3,120899	3,941006	2,983759	2,576759	3,780692	3,679091	2,463911	5,190476
2,16387	2,643341	2,511724	2,135467	0,814835	2,694347	3,24439	2,010801	2,87157	0,48795	2,544053	5,374104
3,941006	2,853319	4,857067	4,704848	2,192542	2,121365	3,754915	2,67724	0,546437	2,107326	1,105646	4,704848
2,135467	4,082649	1,495878	6,134751	1,8133	1,638897	0,468529	2,67724	3,141083	5,327017	2,265442	2,355362
4,025056	3,265548	1,301893	2,495697	3,654302	4,025056	3,351934	7,973866	2,370621	1,380649	1,762566	3,859605
1,943622	3,24439	2,107326	1,675543	2,527842	2,355362	1,626764	3,996735	4,358952	6,197838	2,236068	2,448151
1,472561	4,39155	2,295111	2,432477	3,464102	3,557457	3,351934	1,590607	1,566699	2,038076	3,833032	3,464102



s25	s26	s27	s28	s29	s30	s31	s32	s33	s34	s35	s36
3,351934	5,783097	2,889946	7	4,201829	1,956945	1,997251	3,022225	4,082649	3,223402	2,799291	2,626545
1,903979	1,700186	3,002923	2,310058	2,010801	3,629749	4,232501	1,903979	5,01857	1,638897	2,527842	2,355362
8,334235	6,328319	3,265548	2,340182	3,396223	2,87157	3,351934	2,121365	2,694347	2,609851	2,432477	2,310058
2,853319	4,560954	2,87157	2,67724	3,996735	4,025056	1,970324	2,135467	3,780692	1,50759	3,654302	2,527842
2,107326	3,080988	5,783097	4,141553	3,557457	2,401383	4,111933	3,605427	4,201829	2,295111	4,39155	3,886455
1,414868	1,590607	4,936592	5,422019	2,463911	4,77986	1,290762	6,012611	3,629749	2,265442	3,780692	3,080988
1,392023	2,265442	3,223402	6,535735	4,326757	1,369306	3,061255	2,135467	9,359902	2,463911	1,838956	4,171516
2,295111	2,817181	2,325081	2,527842	3,24439	1,638897	2,401383	1,762566	1,578634	2,024409	2,511724	1,890871
3,886455	4,053692	4,232501	2,817181	1,787838	1,917139	2,76386	3,223402	0,784517	2,908448	4,201829	1,917139
1,213625	1,40343	2,660238	1,838956	3,202579	2,694347	3,373984	1,063124	3,396223	4,082649	0,794607	1,614672
5,01857	3,780692	1,663286	2,609851	3,120899	1,554802	1,877815	3,223402	4,742087	2,626545	3,022225	2,370621
4,936592	4,201829	1,542944	1,943622	2,355362	1,787838	3,061255	1,495878	4,491867	4,457998	2,325081	3,080988
6,134751	3,859605	2,67724	4,491867	5,728539	2,325081	4,491867	1,687843	5,103146	3,120899	1,65107	1,031466
4,560954	3,533802	1,614672	1,116334	1,554802	5,327017	2,889946	3,100868	1,42634	3,265548	4,631914	1,578634
3,654302	2,149636	1,590607	1,687843	3,654302	4,742087	3,080988	2,192542	3,022225	2,817181	4,326757	3,120899
5,422019	3,996735	2,038076	2,87157	2,576759	2,711561	3,373984	2,853319	3,161423	1,472561	2,149636	3,629749
2,87157	1,8133	6,395833	5,422019	4,171516	2,093352	2,835189	4,232501	3,120899	4,596194	5,622535	2,835189
3,022225	2,817181	3,581331	4,668128	2,448151	4,526183	1,449382	2,149636	3,396223	3,96872	3,24439	2,511724
1,290762	10,32307	4,141553	1,531122	2,781518	1,472561	2,889946	2,711561	2,626545	1,40343	1,851859	5,422019
1,449382	2,093352	3,308384	7,172067	2,927078	2,560358	4,936592	2,065591	1,997251	7,547773	1,566699	1,687843



s37	s38	s39	s40	s41	s42	s43	s44	s45	s46	s47	s48
1,50759	2,07944	3,223402	3,833032	3,24439	2,432477	3,396223	3,806729	3,729392	1,460954	2,983759	4,053692
6,197838	7,35429	3,161423	1,903979	4,560954	3,886455	2,964732	5,327017	3,754915	2,983759	2,250719	1,983759
1	2,964732	4,936592	3,441278	4,263541	2,416888	3,041669	2,385961	2,236068	2,927078	1,40343	0,497674
1,930353	3,120899	1,602619	1,224563	3,913586	1,290762	3,806729	2,67724	5,235228	1,137781	6,262341	1,890871
3,629749	4,201829	2,746316	1,460954	5,235228	3,202579	2,192542	3,24439	1,930353	1,224563	3,629749	4,201829
3,161423	1,170134	2,340182	4,39155	2,908448	3,351934	2,964732	3,373984	3,806729	3,330068	1,65107	4,596194
2,927078	3,080988	2,67724	4,082649	3,441278	2,432477	3,487128	2,728883	2,065591	4,326757	1,268584	3,308384
5,895514	1,663286	1,762566	3,24439	2,560358	2,236068	3,654302	2,121365	2,16387	0,855485	0,804714	3,161423
3,351934	2,643341	3,100868	1,983759	3,629749	3,557457	3,24439	2,265442	2,065591	1,554802	3,041669	3,96872
1,246519	4,668128	7,64913	2,20698	2,853319	4,111933	2,416888	2,448151	3,141083	3,859605	2,385961	4,526183
2,236068	2,236068	2,340182	4,818182	1,437844	3,629749	5,01857	1,877815	1,983759	2,121365	1,213625	3,704119
1,983759	8,883301	1,590607	2,401383	4,457998	4,896532	3,533802	2,135467	2,010801	6,395833	4,171516	5,520447
1,75	2,463911	3,859605	1,519338	1,626764	5,622535	6,262341	0,634874	2,250719	3,859605	4,896532	2,817181
4,263541	3,533802	5,327017	2,983759	1,8133	3,96872	3,161423	1,917139	4,171516	1,983759	4,326757	2,853319
3,859605	5,783097	1,917139	4,896532	2,051803	2,250719	2,401383	2,385961	1,826104	3,913586	2,927078	10,54839
2,609851	3,654302	1,712573	2,192542	2,07944	1,346714	2,609851	2,340182	2,385961	1,725004	1,50759	2,051803
3,330068	2,781518	2,432477	2,149636	2,121365	4,082649	2,401383	2,149636	1,602619	3,141083	3,002923	3,141083
2,67724	4,358952	4,025056	3,396223	2,16387	7,861974	2,781518	1,031466	1,626764	2,370621	2,694347	2,051803
3,141083	1,851859	2,511724	2,051803	3,120899	4,171516	4,263541	4,977266	2,927078	3,308384	3,24439	4,326757
2,448151	4,668128	1,800545	2,051803	2,385961	3,441278	3,51036	2,340182	2,192542	1,737479	3,265548	3,141083



s49	s50	s51	s52	s53	s54	s55	s56	s57	s58	s59	s60	s61	s62	s63
3,330068	6,262341	3,533802	1,191828	1,08434	3,373984	1,301893	2,728883	3,441278	1,877815	2,250719	1,65107	3,557457	2,448151	2,835189
1,301893	1,983759	3,464102	1,826104	2,560358	4,201829	2,781518	2,660238	2,609851	1,970324	4,232501	1,838956	4,232501	5,470791	1,712573
2,889946	1,380649	2,853319	2,817181	1,40343	1,775179	1,010468	3,080988	1,675543	2,626545	2,010801	2,728883	2,107326	4,111933	5,374104
2,983759	5,01857	1,554802	2,010801	2,817181	2,340182	3,061255	0,585631	4,424562	3,806729	1,762566	1,983759	3,581331	1,301893	1,566699
3,223402	4,818182	7,35429	2,945839	3,654302	4,141553	4,232501	0,896406	3,061255	1,983759	3,100868	2,853319	5,728539	1,687843	5,953456
3,041669	1,800545	5,01857	2,746316	2,945839	4,77986	2,010801	4,704848	2,527842	3,654302	4,596194	1,614672	1,864812	1,519338	2,67724
4,053692	2,178172	3,041669	1,08434	2,660238	2,051803	2,051803	2,192542	2,660238	2,192542	2,121365	4,232501	3,181921	2,495697	2,121365
1,42634	2,983759	2,560358	4,025056	2,07944	3,913586	2,463911	9,533789	3,351934	1,851859	2,728883	5,190476	2,495697	4,171516	3,96872
2,416888	1,775179	2,093352	4,596194	3,704119	2,325081	5,327017	4,936592	3,202579	1,578634	0,674484	5,520447	1,105646	4,358952	2,560358
3,120899	3,679091	5,190476	1,65107	2,660238	3,24439	3,654302	3,308384	4,596194	3,24439	1,191828	2,385961	2,16387	2,340182	4,818182
4,39155	4,294957	1,638897	2,964732	2,560358	2,024409	2,835189	2,280239	4,201829	3,041669	10,10998	1,566699	0,835127	4,857067	1,554802
2,16387	5,422019	1,930353	2,945839	0,674484	2,889946	4,896532	5,953456	2,781518	1,42634	3,941006	0,855485	2,448151	1,903979	3,605427
7,973866	2,495697	2,609851	3,265548	4,424562	4,818182	1,290762	1,554802	5,190476	5,895514	0,478235	1,712573	7,64913	3,100868	2,051803
4,39155	3,351934	2,051803	1,737479	3,557457	3,654302	4,896532	1,712573	0,458831	4,171516	2,192542	2,093352	3,729392	5,060524	2,010801
2,20698	1,554802	3,581331	2,576759	2,544053	1,191828	2,401383	1,917139	1,380649	2,448151	3,308384	2,576759	1,460954	2,511724	2,121365
1,566699	2,67724	2,07944	2,135467	4,896532	2,576759	4,526183	3,181921	3,679091	1,762566	3,120899	3,080988	2,355362	1,519338	2,07944
3,441278	5,422019	2,781518	1,917139	1,997251	4,631914	4,977266	3,330068	3,581331	3,161423	3,441278	3,330068	2,817181	2,121365	3,022225
2,799291	1,983759	2,67724	2,694347	7,753799	2,463911	2,355362	2,051803	2,728883	3,886455	1,301893	1,301893	4,424562	3,202579	3,704119
2,626545	3,679091	1,725004	2,121365	2,192542	2,983759	2,660238	1,170134	4,704848	2,799291	0,896406	2,889946	3,308384	3,487128	3,833032
3,100868	5,235228	1,890871	3,833032	2,463911	3,351934	2,093352	2,385961	5,190476	2,432477	2,835189	2,67724	0,73429	3,330068	7,172067

G. Source Code To Write Rules To XML

```
public void ruleyaz()
{
    XmlTextReader reader = new XmlTextReader("RulesXML.xml");
    TreeNode tnmodules = new TreeNode();
    tnmodules.Text = "modules";
    treeView1.Nodes.Add(tnmodules);
    TreeNode tnmodule = null;
    TreeNode tnprereq = null;

    while (reader.Read())
    {
      if (reader.NodeType == XmlNodeType.Element && reader.Name ==
"module")
      {
          tnmodule = new TreeNode();
          tnmodule.Text = reader.GetAttribute("mno");
          row = Convert.ToInt32(tnmodule.Text);
          tnmodules.Nodes.Add(tnmodule);
      }
      else if (reader.NodeType == XmlNodeType.Element && reader.Name
== "prereq")
      {
          reader.Read();
          tnprereq = new TreeNode();
          tnprereq.Text = reader.Value;
          col = Convert.ToInt32(tnprereq.Text);
          tnmodule.Nodes.Add(tnprereq);
          prereqs[row, col] = 1;
       }
     }
     reader.Close();
}
```

H. Comparison of Number of Reliable Solutions for the Training and Control Data

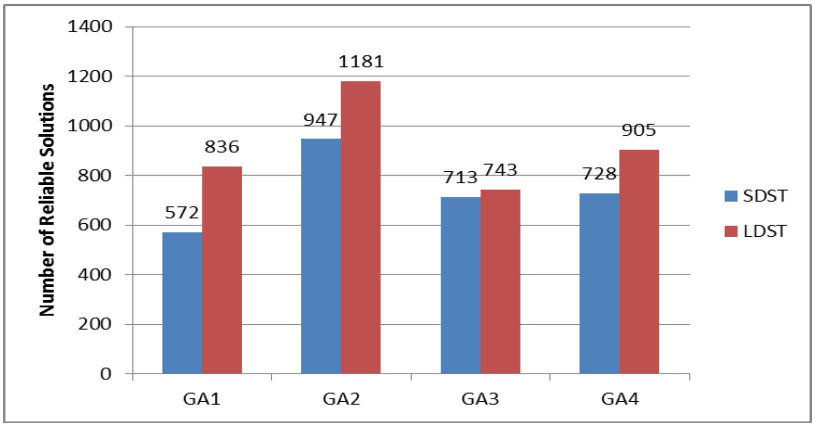


Figure H.1(a) – Comparison of number of reliable solutions for the training data of 100 individuals.

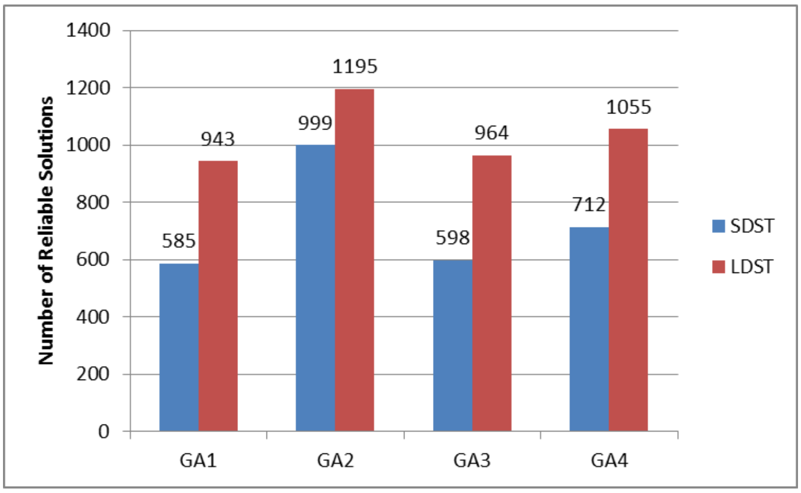


Figure H.1(b) – Comparison of number of reliable solutions for the training data of 120 individuals.

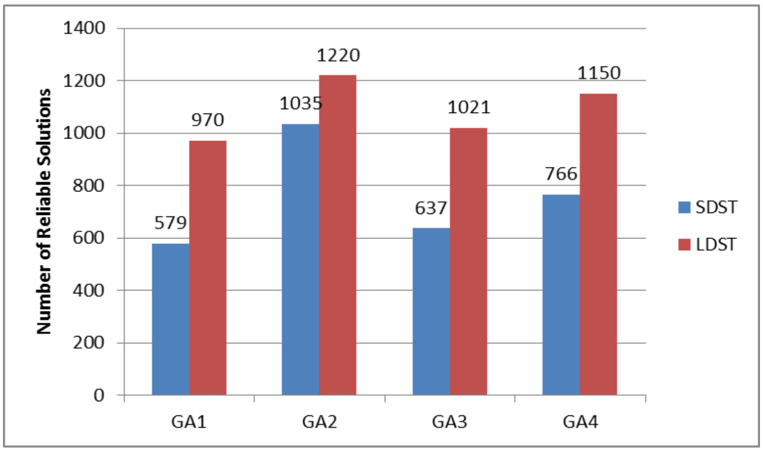


Figure H.1(c) – Comparison of number of reliable solutions for the training data of 140 individuals.

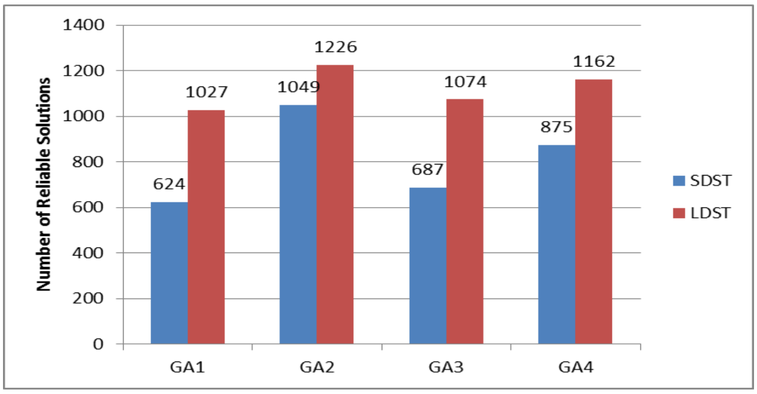


Figure H.1(d) – Comparison of number of reliable solutions for the training data of 160 individuals.

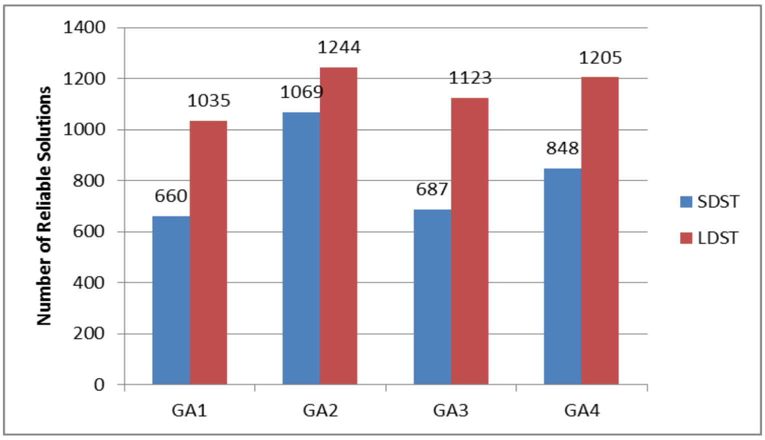


Figure H.1(e) – Comparison of number of reliable solutions for the training data of 180 individuals.

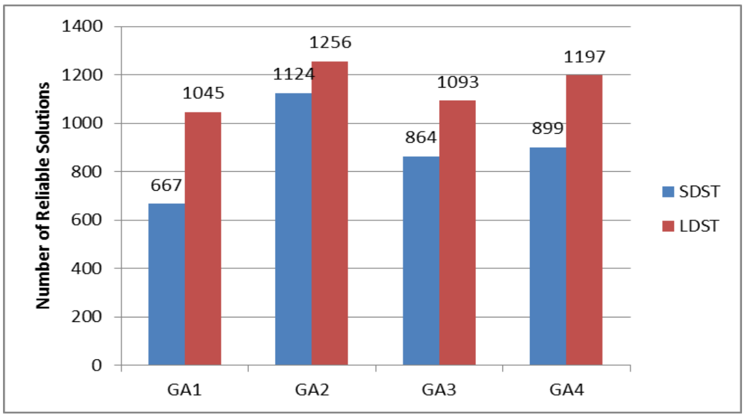


Figure H.1(f) – Comparison of number of reliable solutions for the training data of 200 individuals.

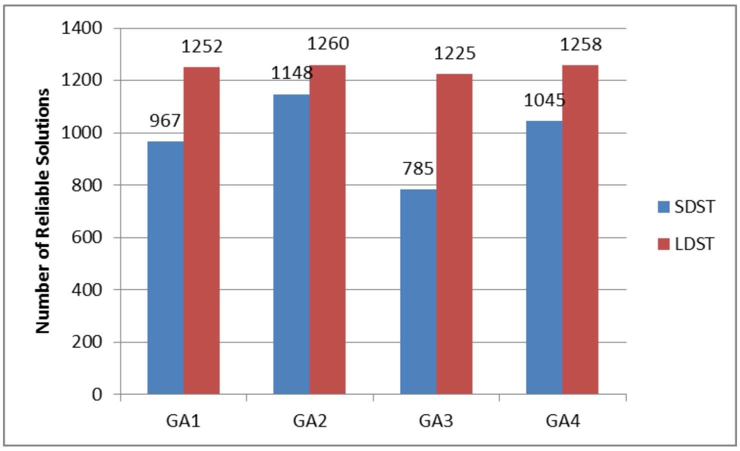


Figure H.2(a) – Comparison of number of reliable solutions for the control data of 100 individuals.

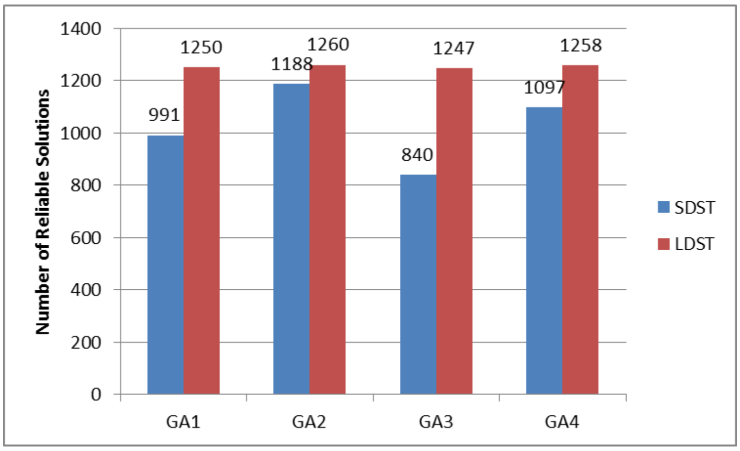


Figure H.2(b) – Comparison of number of reliable solutions for the control data of 120 individuals.

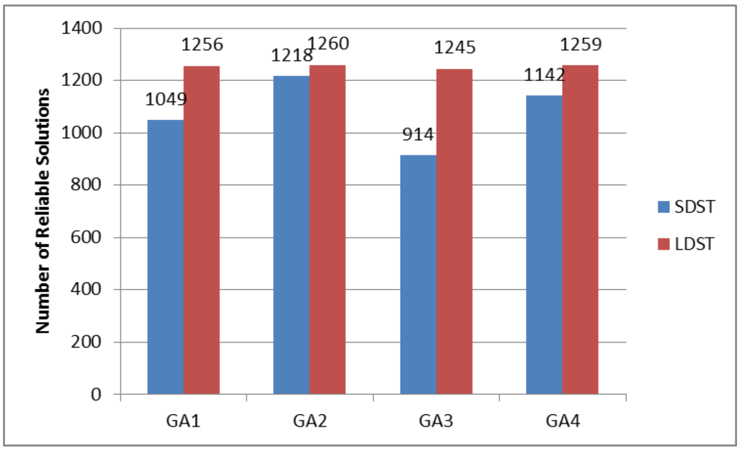


Figure H.2(c) – Comparison of number of reliable solutions for the control data of 140 individuals.

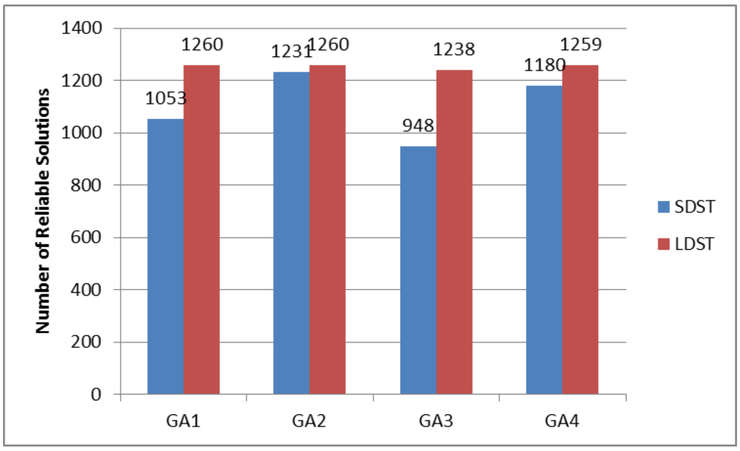


Figure H.2(d) – Comparison of number of reliable solutions for the control data of 160 individuals.

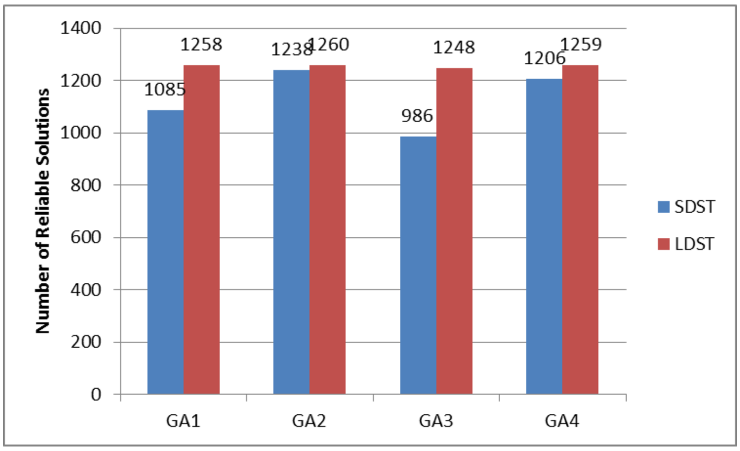


Figure H.2(e) – Comparison of number of reliable solutions for the control data of 180 individuals.

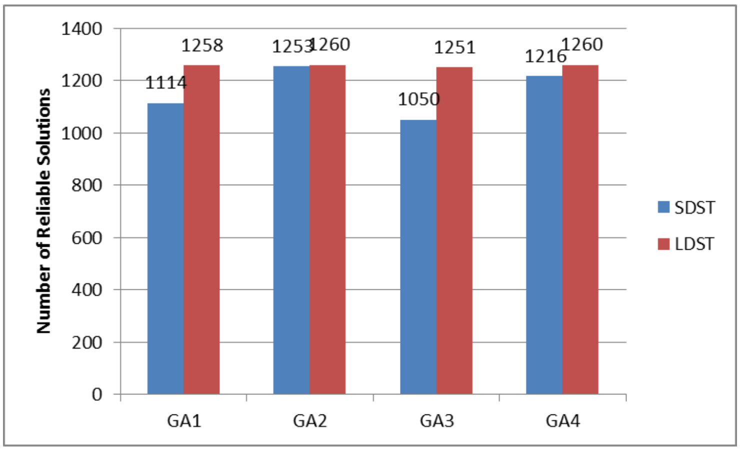


Figure H.2(f) – Comparison of number of reliable solutions for the control data of 200 individuals.

İ. THE GRAPHICS OF THE NUMBER OF RELIABLE MODULE RANGES FOR TRAINING AND CONTROL DATA

Training Data

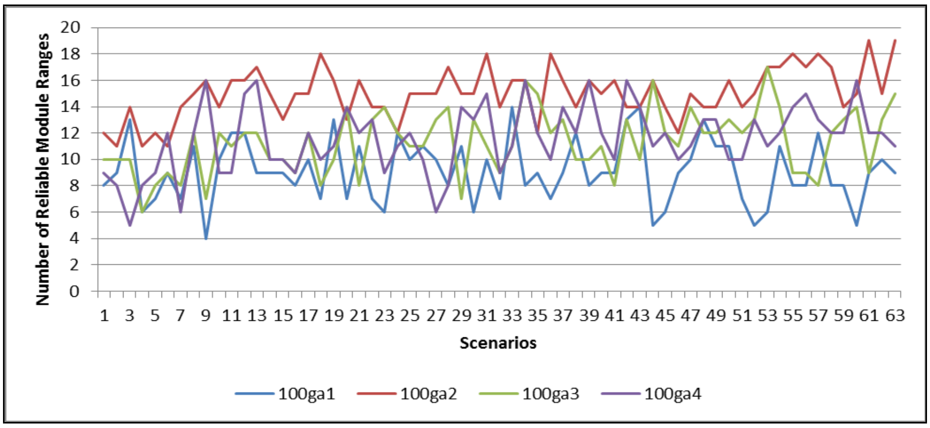


Figure I.1(a) – Number of reliable module ranges in training data – SDST for a population size of 100.

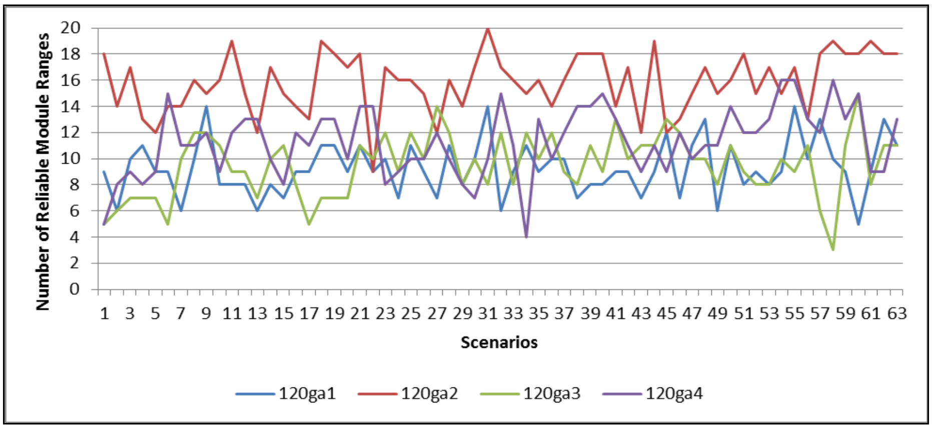


Figure I.1(b) – Number of reliable module ranges in training data – SDST for a population size of 120.

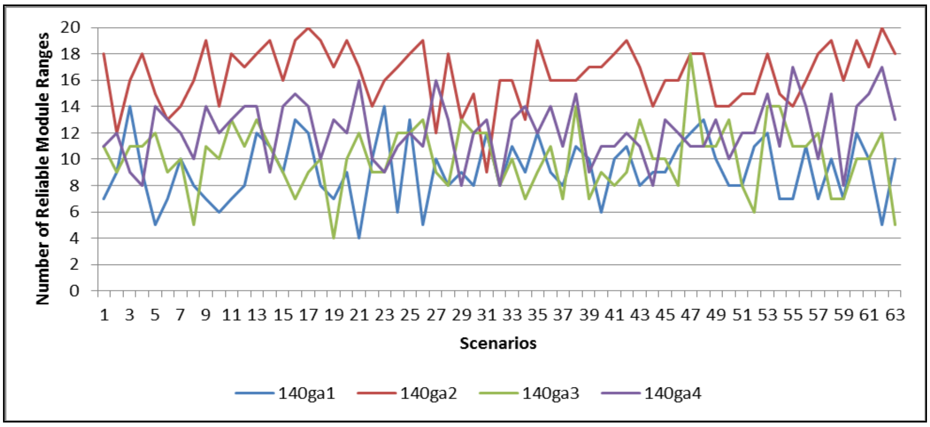


Figure I.1(c) – Number of reliable module ranges in training data – SDST for a population size of 140.

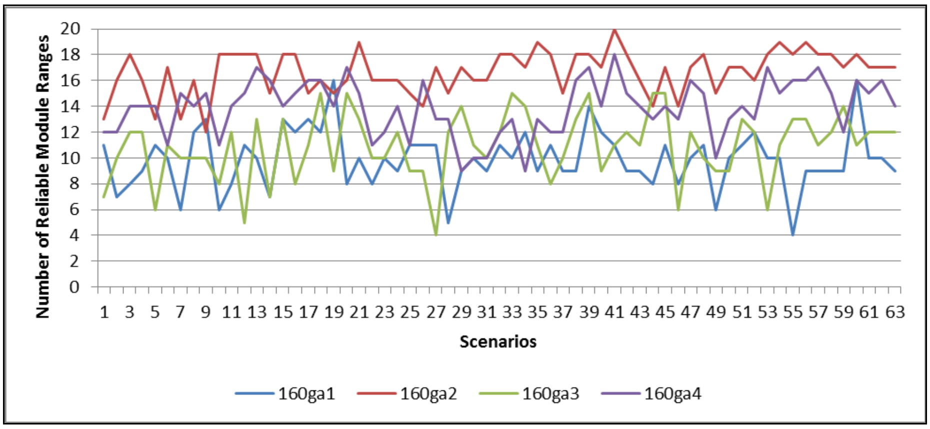


Figure I.1(d) – Number of reliable module ranges in training data – SDST for a population size of 160.

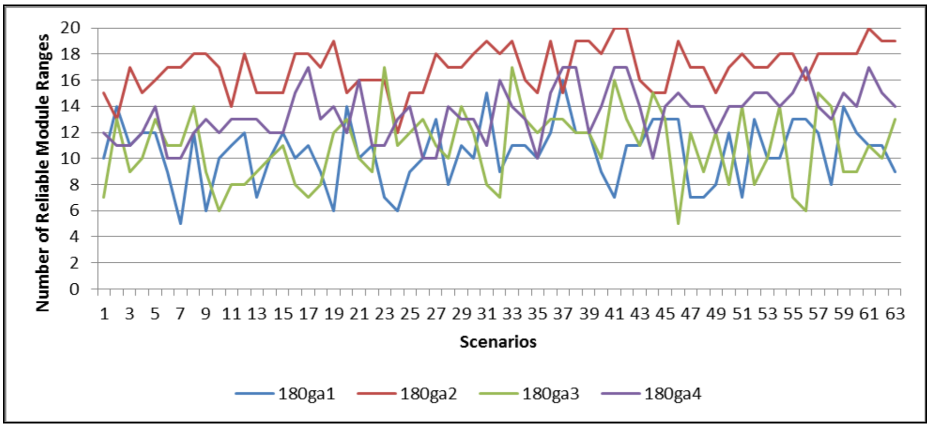


Figure I.1(e) – Number of reliable module ranges in training data – SDST for a population size of 180.

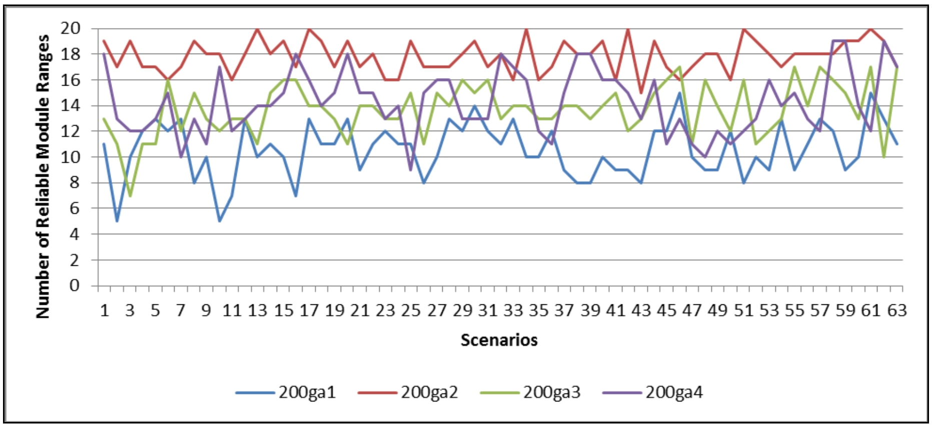


Figure I.1(f) – Number of reliable module ranges in training data – SDST for a population size of 200.

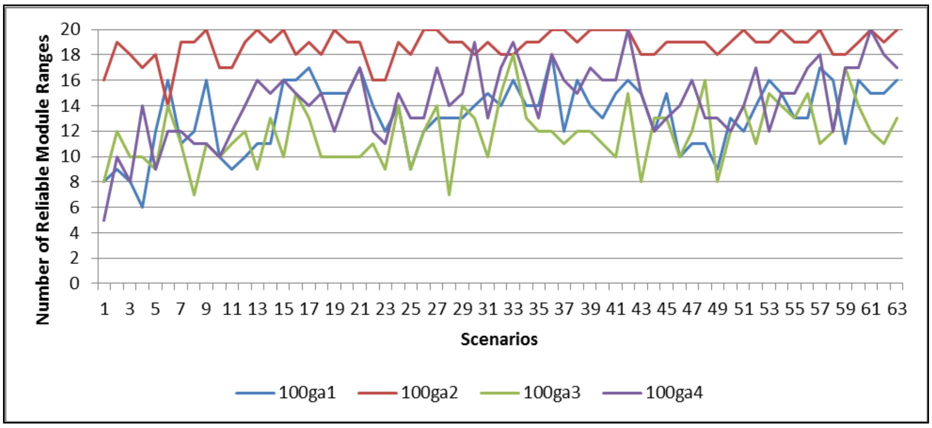


Figure I.2(a) – Number of reliable module ranges in training data – LDST for a population size of 100.

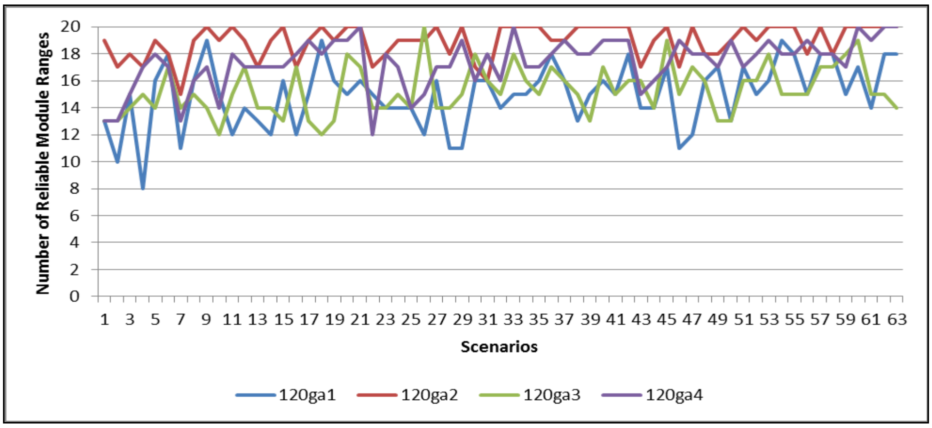


Figure I.2(b) – Number of reliable module ranges in training data – LDST for a population size of 120.

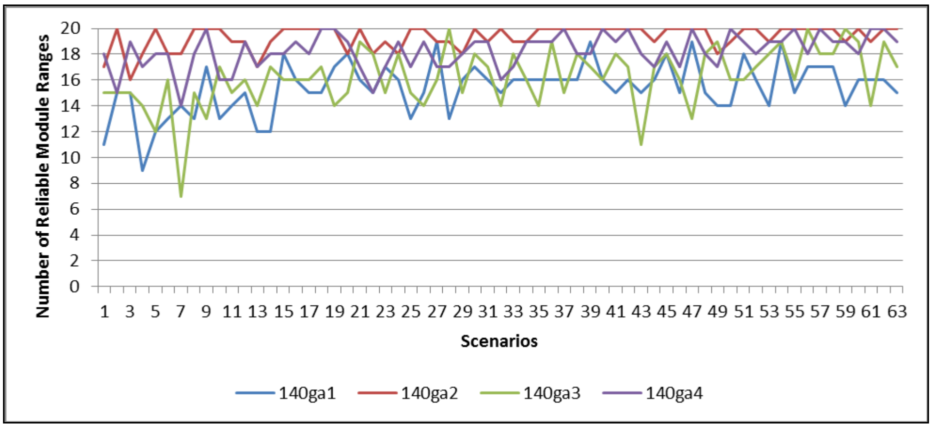


Figure I.2(c) – Number of reliable module ranges in training data – LDST for a population size of 140.

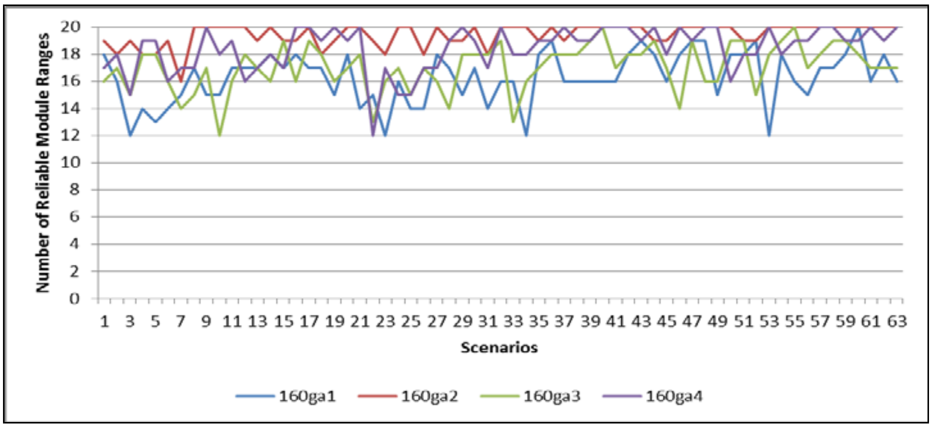


Figure I.2(d) – Number of reliable module ranges in training data – LDST for a population size of 160.

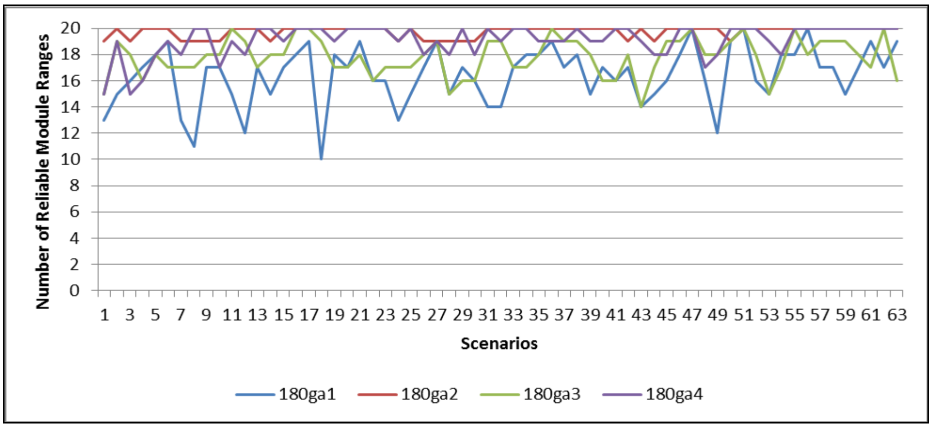


Figure I.2(e) – Number of reliable module ranges in training data – LDST for a population size of 180.

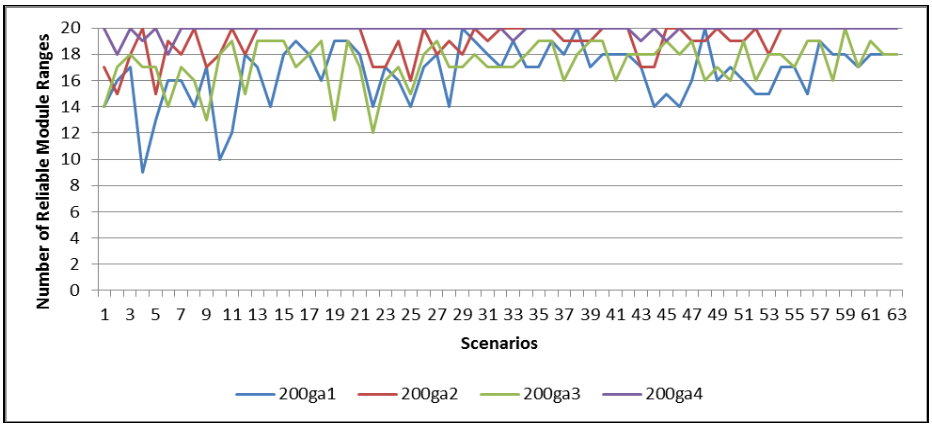


Figure I.2(f) – Number of reliable module ranges in training data – LDST for a population size of 200.

Control Data

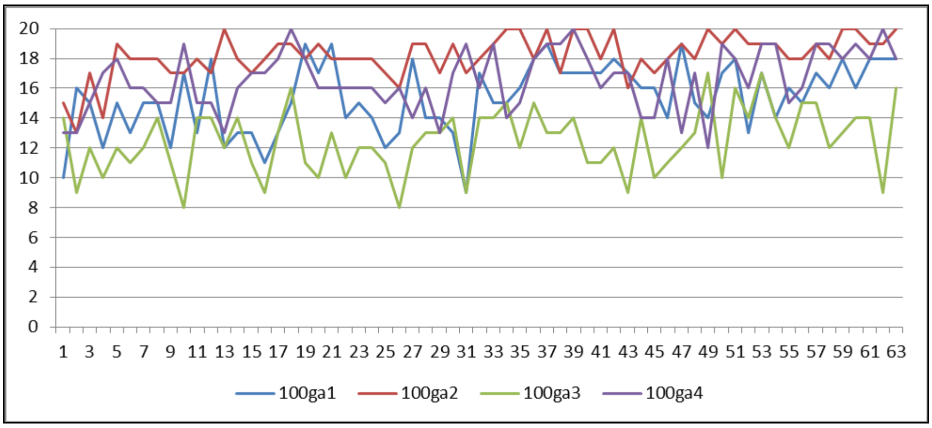


Figure I.3(a) – Number of reliable module ranges in control data – SDST for a population size of 100.

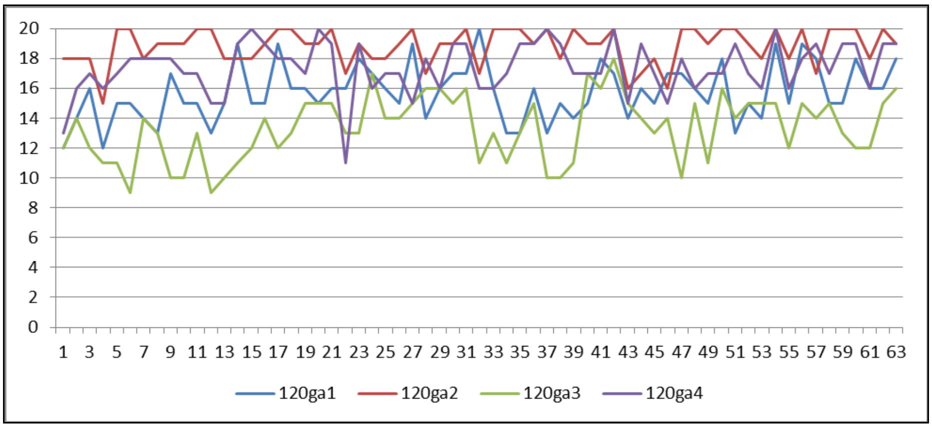


Figure I.3(b) – Number of reliable module ranges in control data – SDST for a population size of 120.

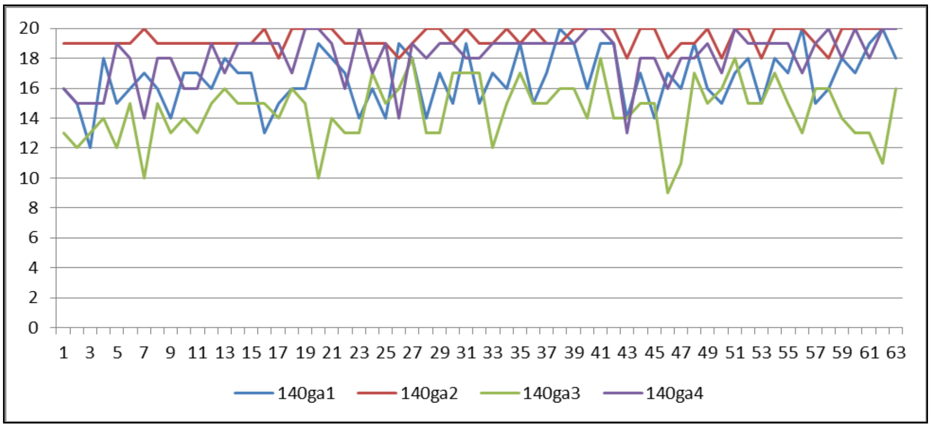


Figure I.3(c) – Number of reliable module ranges in control data – SDST for a population size of 140.

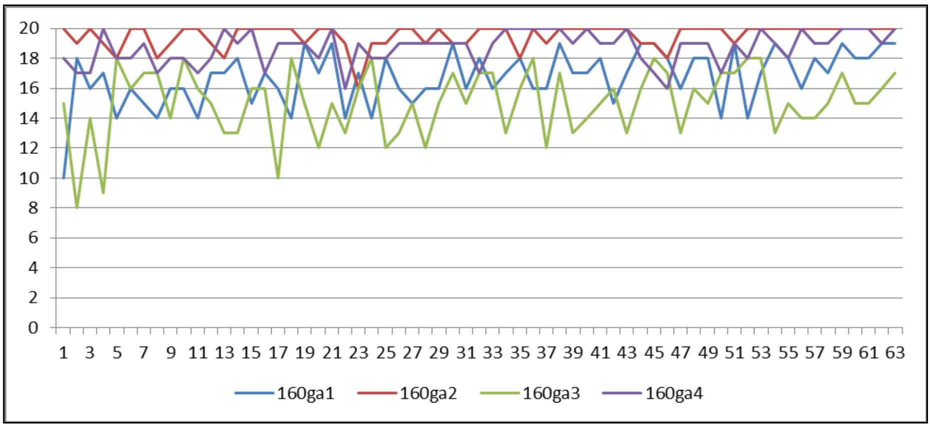


Figure I.3(d) – Number of reliable module ranges in control data – SDST for a population size of 160.

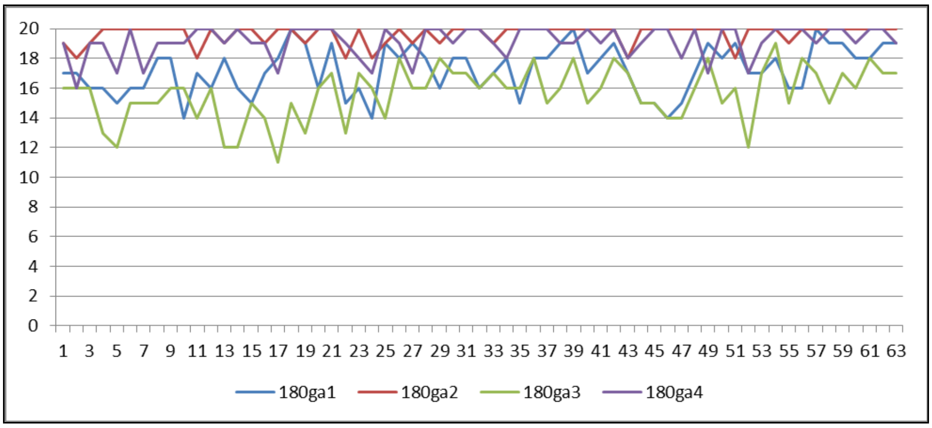


Figure I.3(e) – Number of reliable module ranges in control data – SDST for a population size of 180.

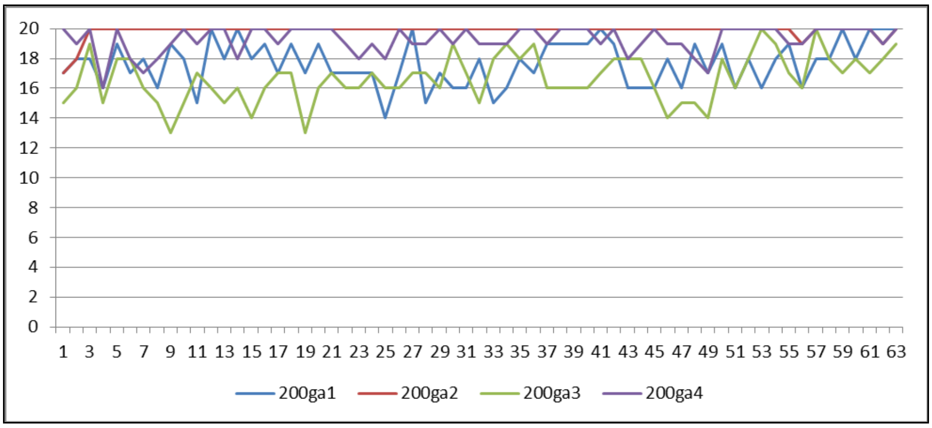


Figure I.3(f) – Number of reliable module ranges in control data – SDST for a population size of 200.

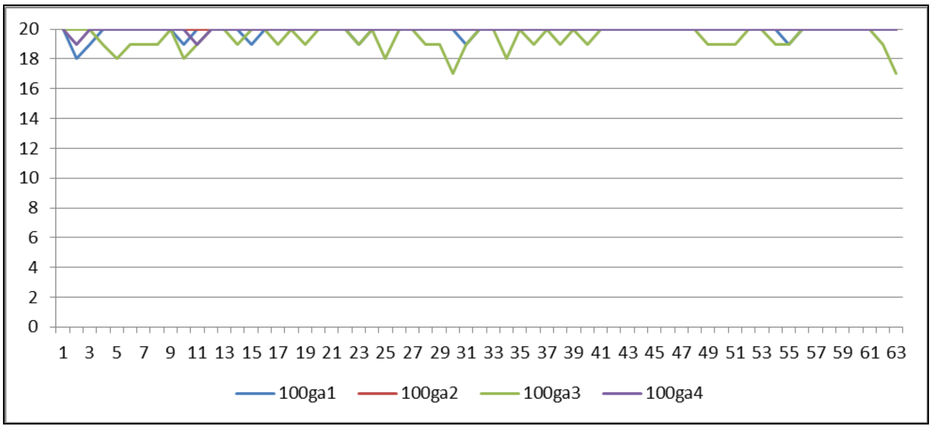


Figure I.4(a) – Number of reliable module ranges in control data – LDST for a population size of 100.

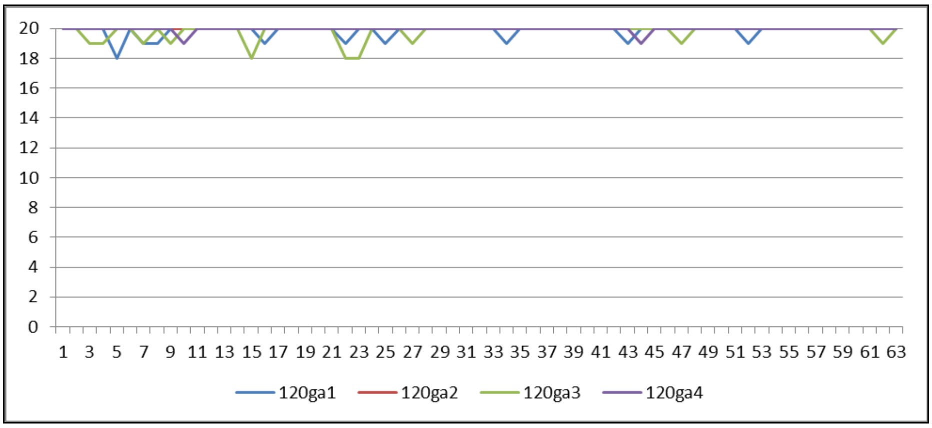


Figure I.4(b) – Number of reliable module ranges in control data – LDST for a population size of 120.

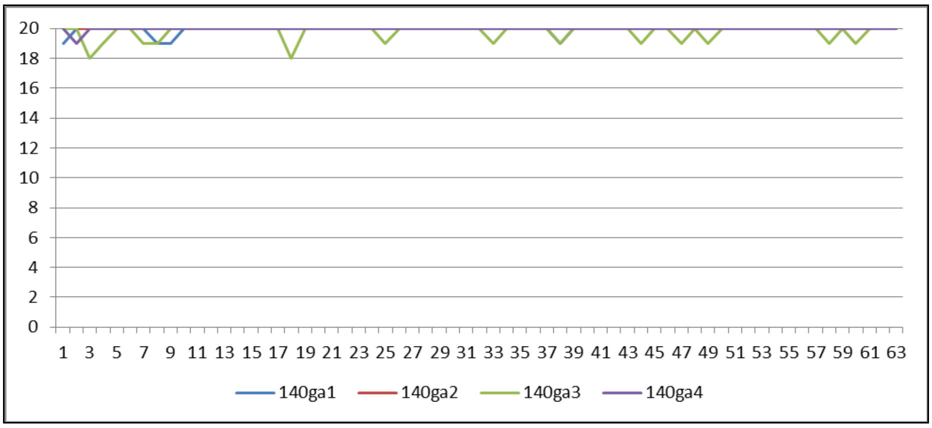


Figure I.4(c) – Number of reliable module ranges in control data – LDST for a population size of 140.

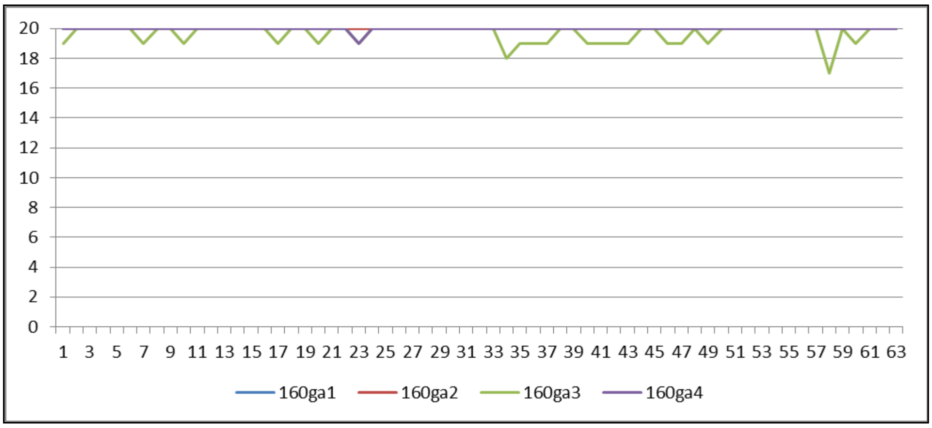


Figure I.4(d) – Number of reliable module ranges in control data – LDST for a population size of 160.

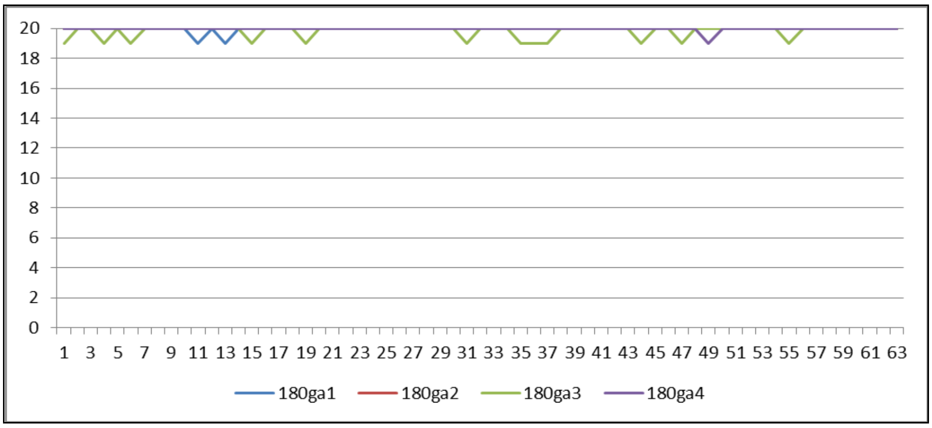


Figure I.4(e) – Number of reliable module ranges in control data – LDST for a population size of 180.

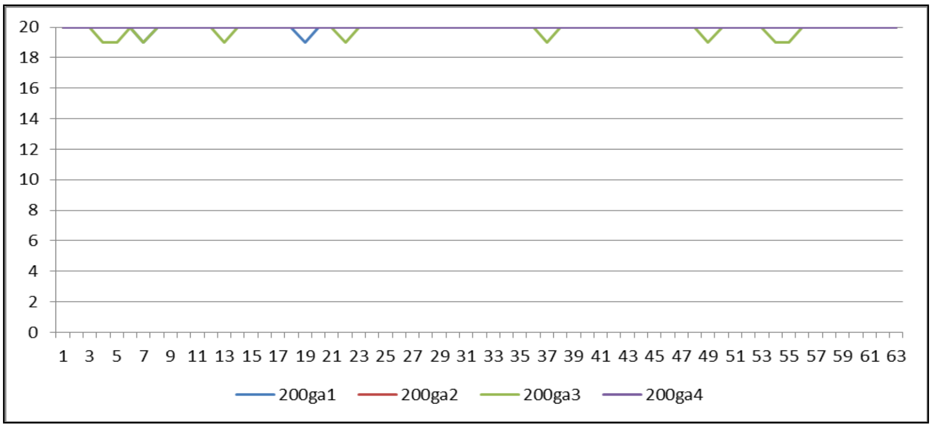


Figure I.4(f) – Number of reliable module ranges in control data – LDST for a population size of 200.

J. BEST MODULE RANGES FOR THE TRAINING DATA

Table J.1(a) – Best module ranges for SDST and 100 individuals.

OX	OX2	PMX	OX’
4	1	4	1
1	4	1	4
8	8	8	9
9	9	6	6
10	10	9	8
13	13	10	10
14	6	13	14
6	14	14	13
15	15	15	15
16	16	26	16
20	20	16	23
29	23	20	26
23	29	32	29
35	26	23	20
26	32	35	32
38	35	29	35
32	38	38	38

Table J.1(b) – Best module ranges for SDST and 120 individuals.

OX	OX2	PMX	OX’
1	4	1	4
4	1	4	1
10	8	10	6
8	6	6	9
6	9	8	10
9	10	9	8
14	14	14	13
13	13	13	14
15	15	15	16
16	16	16	15
20	26	20	26
29	20	29	23
23	23	23	20
38	35	35	35
35	32	32	38
32	29	26	29
26	38	38	32

Table J.1(c) – Best module ranges for SDST and 140 individuals.

OX	OX2	PMX	OX’
4	4	1	1
1	1	4	4
8	8	8	9
9	9	6	8
10	10	13	6
13	6	9	10
14	13	15	14
6	14	14	13
16	16	16	16
23	15	10	15
15	20	20	23
26	29	23	20
20	26	29	26
29	23	26	29
32	32	32	35
38	38	35	32
35	35	38	38

Table J.1(d) – Best module ranges for SDST and 160 individuals.

OX	OX2	PMX	OX’
1	1	1	4
4	4	4	1
6	8	6	8
9	6	8	10
8	9	9	6
10	10	13	13
14	14	26	9
13	13	10	14
16	15	14	15
23	16	16	16
15	23	23	23
20	20	15	20
29	26	20	26
38	32	29	29
32	35	32	32
26	38	38	35
35	29	35	38

Table J.1(e) – Best module ranges for SDST and 180 individuals.

OX	OX2	PMX	OX’
1	1	4	1
4	4	1	4
6	6	6	6
8	9	9	8
9	8	8	9
10	10	14	13
13	13	13	10
26	14	10	14
14	16	15	26
15	15	16	16
16	20	20	23
20	23	23	15
32	26	26	20
23	29	29	35
29	38	32	32
38	35	35	29
35	32	38	38

Table J.1(f) – Best module ranges for SDST and 200 individuals.

OX	OX2	PMX	OX’
4	4	1	1
1	1	4	4
6	6	9	9
8	8	8	8
13	9	10	6
9	10	14	10
10	14	6	14
14	13	13	13
16	16	15	15
23	15	16	16
15	26	29	20
32	23	20	23
26	20	26	26
20	29	23	38
38	32	38	32
29	35	35	29
35	38	32	35

Table J.2(a) – Best module ranges for LDST and 100 individuals.

OX	OX2	PMX	OX’
10	1	1	1
1	2	2	2
2	4	4	4
4	6	6	8
9	5	10	13
11	9	5	5
12	10	12	9
5	8	17	10
8	11	8	6
14	12	26	11
13	17	13	21
6	14	9	14
17	13	11	25
15	16	25	12
20	23	15	16
21	20	21	20
16	21	14	15
32	15	16	17
23	31	23	23
31	22	20	31
24	32	31	29
33	24	22	26
26	25	32	32
29	26	38	22
22	33	33	38
25	29	35	34
34	34	29	24
36	36	34	35
38	35	24	36
35	38	36	33

Table J.2 (b) – Best module ranges for LDST and 120 individuals.

OX	OX2	PMX	OX’
8	4	4	1
1	5	1	4
4	1	10	9
2	2	2	2
9	8	8	10
17	10	11	8
14	9	14	12
6	11	9	13
11	14	6	11
20	6	12	5
10	12	15	14
5	13	5	17
15	16	17	16
12	17	23	6
13	23	29	23
21	25	20	20
22	15	13	21
25	20	35	15
16	22	16	26
23	24	21	29
26	29	26	22
31	21	22	34
29	31	31	24
32	32	24	25
33	33	25	35
24	34	34	31
38	26	36	36
35	36	38	38
34	38	32	32
36	35	33	33

Table J.2(c) – Best module ranges for LDST and 140 individuals.

OX	OX2	PMX	OX’
4	4	1	5
1	1	8	1
2	2	2	2
17	5	9	4
9	6	5	8
8	9	10	10
10	8	15	9
5	10	4	6
11	12	17	12
6	13	12	14
12	11	20	13
13	14	14	15
14	15	13	16
21	16	25	11
20	17	6	17
23	23	11	21
29	20	22	20
31	21	21	22
15	22	16	32
32	26	23	23
26	29	26	29
16	34	31	35
22	24	34	26
24	25	29	34
38	35	35	38
35	38	36	36
33	31	32	31
25	33	33	24
34	32	38	25
36	36	24	33

Table J.2(d) – Best module ranges for LDST and 160 individuals.

OX	OX2	PMX	OX’
5	4	1	1
1	1	2	4
4	2	4	2
9	5	5	6
2	6	9	8
6	8	8	5
10	9	6	10
14	10	11	12
12	12	12	13
8	11	17	11
13	14	29	9
11	13	10	14
16	16	14	15
20	26	13	16
17	17	16	17
24	21	15	21
22	15	25	26
21	23	26	23
26	20	23	25
29	24	21	20
34	25	20	22
15	31	22	29
25	32	24	34
23	22	35	31
36	33	31	32
31	35	34	38
32	38	36	36
33	29	38	33
38	34	33	24
35	36	32	35

Table J.2(e) – Best module ranges for LDST and 180 individuals.

OX	OX2	PMX	OX’
4	4	4	1
2	1	1	4
1	2	2	2
8	8	9	9
5	12	8	10
9	6	12	6
10	5	11	12
14	9	13	8
6	10	6	11
12	13	5	5
13	11	14	14
15	14	10	13
11	16	21	16
16	17	16	15
17	21	15	17
20	29	26	21
22	23	17	26
21	26	29	23
26	15	23	29
23	20	31	20
38	24	20	24
31	22	38	38
24	31	22	22
32	25	35	31
29	34	24	35
35	36	32	32
34	32	25	25
25	33	34	33
33	38	36	34
36	35	33	36

Table J.2(f) – Best module ranges for LDST and 200 individuals.

OX	OX2	PMX	OX’
1	1	1	1
2	2	4	2
9	4	8	4
4	5	2	5
6	8	5	9
8	9	9	6
10	6	12	10
11	10	6	8
12	14	10	14
14	12	13	12
13	13	17	13
17	16	14	16
5	20	16	11
16	17	11	17
21	11	15	15
22	15	29	23
15	21	23	20
23	23	22	26
20	24	26	24
26	29	20	21
24	22	38	22
29	35	24	25
25	25	21	33
34	38	31	29
31	26	35	38
32	31	32	31
35	32	25	32
33	33	34	34
38	34	36	35
36	36	33	36

K. Best Module Ranges for the Control Data

Table K.1(a) – Best module ranges for SDST and 100 individuals.

OX	OX2	PMX	OX’
1	1	1	1
3	5	4	3
4	4	3	4
5	9	5	5
11	3	11	7
7	7	7	9
12	11	12	11
9	12	9	12
14	14	14	14
19	19	19	29
29	29	30	19
22	21	29	21
34	22	22	22
30	30	21	34
21	34	40	36
36	40	36	30
40	36	34	40

Table K.1(b) – Best module ranges for SDST and 120 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	4
3	5	3	3
11	3	5	5
5	7	11	14
7	11	7	7
12	12	14	11
14	9	12	9
9	14	9	12
19	19	19	19
22	22	29	21
21	21	21	22
29	29	34	29
30	34	36	30
36	30	22	36
34	36	30	34
40	40	40	40

Table K.1(c) – Best module ranges for SDST and 140 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	3
3	5	3	4
7	3	5	5
5	11	7	7
11	7	11	11
12	9	9	12
9	12	12	9
29	14	14	14
14	19	19	22
19	21	21	19
21	29	29	21
30	22	34	29
22	30	36	30
34	40	22	36
36	36	40	34
40	34	30	40

Table K.1(d) – Best module ranges for SDST and 160 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	4
5	3	3	5
3	5	14	3
7	11	5	7
9	7	7	11
11	12	9	9
12	9	11	12
14	14	12	14
19	19	19	19
21	21	22	22
29	29	21	29
34	22	29	30
22	30	34	21
40	40	30	36
30	36	36	40
36	34	40	34

Table K.1(e) – Best module ranges for SDST and 180 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	3	3
3	3	4	4
9	5	5	5
11	7	11	11
5	9	12	7
7	11	7	9
12	12	14	12
14	14	9	14
19	19	19	19
22	22	29	22
30	21	21	21
29	29	22	29
21	30	36	34
34	40	40	40
36	36	34	30
40	34	30	36

Table K.1(f) – Best module ranges for SDST and 200 individuals.

OX	OX2	PMX	OX’
1	1	1	1
3	4	4	3
4	3	3	4
5	5	5	5
11	7	11	11
14	11	7	7
7	12	12	12
9	14	9	14
12	9	14	22
19	19	19	9
22	29	29	19
21	21	21	21
36	22	22	29
30	36	40	34
29	30	30	30
34	34	36	40
40	40	34	36

Table K.2(a) – Best module ranges for LDST and 100 individuals.

OX	OX2	PMX	OX’
1	1	11	1
4	4	1	3
5	3	3	4
3	5	4	5
7	11	7	7
14	7	5	11
11	14	13	14
18	13	23	16
23	9	17	9
25	16	25	22
13	19	14	19
9	18	18	21
22	23	16	17
19	21	9	18
17	25	29	23
28	22	19	25
16	28	28	13
21	17	21	26
26	31	22	29
29	29	33	30
30	33	26	28
31	26	30	31
33	30	31	33
34	34	34	34
35	35	35	37
37	37	37	38
38	38	38	35
39	39	39	39
41	40	40	40
40	41	41	41

Table K.2 (b) – Best module ranges for LDST and 120 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	4
3	14	14	7
5	3	7	5
11	11	5	3
7	7	16	14
9	5	9	11
14	13	13	19
23	22	17	9
17	9	22	22
16	16	3	16
13	18	18	23
30	19	19	13
31	23	25	17
18	21	26	25
19	17	23	18
25	25	21	26
26	26	11	21
28	30	30	30
29	29	28	31
21	28	31	29
22	31	33	28
33	33	29	33
34	34	34	34
35	35	35	35
37	37	37	37
40	38	38	38
38	39	39	39
39	40	40	40
41	41	41	41

Table K.2(c) – Best module ranges for LDST and 140 individuals.

OX	OX2	PMX	OX’
1	1	1	1
3	4	4	4
4	3	5	3
14	14	3	11
21	11	7	5
5	5	18	7
11	7	17	14
7	9	11	9
9	13	13	16
16	16	14	13
23	17	9	19
22	23	16	18
13	19	19	23
17	25	25	17
18	22	26	33
31	28	33	25
29	18	23	22
19	21	28	21
25	30	21	31
26	29	34	29
30	26	30	30
34	34	31	26
28	31	22	28
33	33	29	34
35	35	35	37
39	37	37	35
37	38	38	38
38	39	39	39
40	40	40	40
41	41	41	41

Table K.2(d) – Best module ranges for LDST and 160 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	4
3	14	5	14
5	3	3	3
14	5	7	5
7	7	11	7
11	11	13	9
17	9	9	11
9	13	14	16
16	23	19	13
13	18	17	19
23	22	22	23
30	19	23	22
31	16	25	25
18	17	16	21
19	21	28	30
25	25	29	18
21	26	21	26
26	30	18	17
28	29	31	29
29	33	26	28
22	31	34	33
33	34	33	31
34	37	30	34
35	28	35	35
37	35	37	37
38	38	38	38
39	39	40	39
40	40	39	40
41	41	41	41

Table K.2(e) – Best module ranges for LDST and 180 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	4
3	5	3	14
14	3	11	5
7	7	5	3
5	17	7	7
11	11	14	11
16	13	19	16
9	14	16	9
17	9	13	13
23	16	9	18
22	19	17	23
13	23	23	25
30	22	22	19
31	21	28	22
18	18	18	17
19	25	30	21
25	28	25	29
26	31	29	31
28	30	31	26
29	26	21	30
33	33	37	28
21	29	26	33
34	34	33	34
35	35	41	37
37	37	34	35
38	38	39	39
40	39	35	38
39	40	38	41
41	41	40	40

Table K.2(f) – Best module ranges for LDST and 200 individuals.

OX	OX2	PMX	OX’
1	1	1	1
4	4	4	4
5	5	14	3
3	3	3	5
7	11	5	7
11	7	11	9
14	9	9	17
19	13	7	11
9	14	18	13
16	16	25	14
13	19	16	16
31	23	19	23
23	21	13	18
22	30	17	25
21	22	23	19
30	25	28	21
17	17	30	22
18	18	21	31
25	28	33	30
26	33	26	29
28	26	29	33
29	29	22	26
33	31	31	28
34	34	34	34
40	37	35	35
35	35	41	37
37	38	37	38
38	39	38	39
39	40	39	40
41	41	40	41

L. Reliable Module Ranges with Reliability Percentages of the Training Data

Table L.1(a) – Reliability of best module ranges of SDST for a population size of 100.

Expert	**OX**	**OX2**	**PMX**	**OX'**
Reliability	93.87%	97.30%	96.56%	**97.55%**
1	4	1	4	1
4	1	4	1	4
6	8	8	8	9
8	9	9	6	6
9	10	10	9	8
10	13	13	10	10
13	14	6	13	14
14	6	14	14	13
15	15	15	15	15
16	16	16	26	16
20	20	20	16	23
23	29	23	20	26
26	23	29	32	29
29	35	26	23	20
32	26	32	35	32
35	38	35	29	35
38	32	38	38	38

Table L.1(b) – Reliability of best module ranges of SDST for a population size of 120.

Expert	**OX**	**OX2**	**PMX**	**OX'**
Reliability	94.11%	**97.55%**	96.07%	95.83%
1	1	4	1	4
4	4	1	4	1
6	10	8	10	6
8	8	6	6	9
9	6	9	8	10
10	9	10	9	8
13	14	14	14	13
14	13	13	13	14
15	15	15	15	16
16	16	16	16	15
20	20	26	20	26
23	29	20	29	23
26	23	23	23	20
29	38	35	35	35
32	35	32	32	38
35	32	29	26	29
38	26	38	38	32

Table L.1(c) – Reliability of best module ranges of SDST for a population size of 140.

Expert	OX	OX2	PMX	OX’
Reliability	94.11%	96.81%	96.32%	**98.04%**
1	4	4	1	1
4	1	1	4	4
6	8	8	8	9
8	9	9	6	8
9	10	10	13	6
10	13	6	9	10
13	14	13	15	14
14	6	14	14	13
15	16	16	16	16
16	23	15	10	15
20	15	20	20	23
23	26	29	23	20
26	20	26	29	26
29	29	23	26	29
32	32	32	32	35
35	38	38	35	32
38	35	35	38	38

Table L.1(d) – Reliability of best module ranges of SDST for a population size of 160.

Expert	OX	OX2	PMX	OX’
Reliability	95.83%	**97.79%**	92.89%	**97.79%**
1	1	1	1	4
4	4	4	4	1
6	6	8	6	8
8	9	6	8	10
9	8	9	9	6
10	10	10	13	13
13	14	14	26	9
14	13	13	10	14
15	16	15	14	15
16	23	16	16	16
20	15	23	23	23
23	20	20	15	20
26	29	26	20	26
29	38	32	29	29
32	32	35	32	32
35	26	38	38	35
38	35	29	35	38

Table L.1(e) – Reliability of best module ranges of SDST for a population size of 180.

Expert	OX	OX2	PMX	OX’
Reliability	95.10%	**98.53%**	**98.53%**	95.10%
1	1	1	4	1
4	4	4	1	4
6	6	6	6	6
8	8	9	9	8
9	9	8	8	9
10	10	10	14	13
13	13	13	13	10
14	26	14	10	14
15	14	16	15	26
16	15	15	16	16
20	16	20	20	23
23	20	23	23	15
26	32	26	26	20
29	23	29	29	35
32	29	38	32	32
35	38	35	35	29
38	35	32	38	38

Table L.1(f) – Reliability of best module ranges of SDST for a population size of 200.

Expert	OX	OX2	PMX	OX’
Reliability	94.60%	**98.28%**	94.11%	97.06%
1	4	4	1	1
4	1	1	4	4
6	6	6	9	9
8	8	8	8	8
9	13	9	10	6
10	9	10	14	10
13	10	14	6	14
14	14	13	13	13
15	16	16	15	15
16	23	15	16	16
20	15	26	29	20
23	32	23	20	23
26	26	20	26	26
29	20	29	23	38
32	38	32	38	32
35	29	35	35	29
38	35	38	32	35

Table L.2(a) – Reliability of best module ranges of LDST for a population size of 100.

Expert	**OX**	**OX2**	**PMX**	**OX'**
Reliability	92.61%	**96.93%**	87.85%	92.12%
1	10	1	1	1
2	1	2	2	2
4	2	4	4	4
5	4	6	6	8
6	9	5	10	13
8	11	9	5	5
9	12	10	12	9
10	5	8	17	10
11	8	11	8	6
12	14	12	26	11
13	13	17	13	21
14	6	14	9	14
15	17	13	11	25
16	15	16	25	12
17	20	23	15	16
20	21	20	21	20
21	16	21	14	15
22	32	15	16	17
23	23	31	23	23
24	31	22	20	31
25	24	32	31	29
26	33	24	22	26
29	26	25	32	32
31	29	26	38	22
32	22	33	33	38
33	25	29	35	34
34	34	34	29	24
35	36	36	34	35
36	38	35	24	36
38	35	38	36	33

Table L.2(b) – Reliability of best module ranges of LDST for a population size of 120.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	91.45%	**95.86%**	88.70%	92.43%
1	8	4	4	1
2	1	5	1	4
4	4	1	10	9
5	2	2	2	2
6	9	8	8	10
8	17	10	11	8
9	14	9	14	12
10	6	11	9	13
11	11	14	6	11
12	20	6	12	5
13	10	12	15	14
14	5	13	5	17
15	15	16	17	16
16	12	17	23	6
17	13	23	29	23
20	21	25	20	20
21	22	15	13	21
22	25	20	35	15
23	16	22	16	26
24	23	24	21	29
25	26	29	26	22
26	31	21	22	34
29	29	31	31	24
31	32	32	24	25
32	33	33	25	35
33	24	34	34	31
34	38	26	36	36
35	35	36	38	38
36	34	38	32	32
38	36	35	33	33

Table L.2(c) – Reliability of best module ranges of LDST for a population size of 140.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	88.47%	**97.06%**	88.16%	92.57%
1	4	4	1	5
2	1	1	8	1
4	2	2	2	2
5	17	5	9	4
6	9	6	5	8
8	8	9	10	10
9	10	8	15	9
10	5	10	4	6
11	11	12	17	12
12	6	13	12	14
13	12	11	20	13
14	13	14	14	15
15	14	15	13	16
16	21	16	25	11
17	20	17	6	17
20	23	23	11	21
21	29	20	22	20
22	31	21	21	22
23	15	22	16	32
24	32	26	23	23
25	26	29	26	29
26	16	34	31	35
29	22	24	34	26
31	24	25	29	34
32	38	35	35	38
33	35	38	36	36
34	33	31	32	31
35	25	33	33	24
36	34	32	38	25
38	36	36	24	33

Table L.2(d) – Reliability of best module ranges of LDST for a population size of 160.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	93.50%	**95.77%**	91.85%	95.19%
1	5	4	1	1
2	1	1	2	4
4	4	2	4	2
5	9	5	5	6
6	2	6	9	8
8	6	8	8	5
9	10	9	6	10
10	14	10	11	12
11	12	12	12	13
12	8	11	17	11
13	13	14	29	9
14	11	13	10	14
15	16	16	14	15
16	20	26	13	16
17	17	17	16	17
20	24	21	15	21
21	22	15	25	26
22	21	23	26	23
23	26	20	23	25
24	29	24	21	20
25	34	25	20	22
26	15	31	22	29
29	25	32	24	34
31	23	22	35	31
32	36	33	31	32
33	31	35	34	38
34	32	38	36	36
35	33	29	38	33
36	38	34	33	24
38	35	36	32	35

Table L.2(e) – Reliability of best module ranges of LDST for a population size of 180.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	94.70%	**94.84%**	90.56%	93.81%
1	4	4	4	1
2	2	1	1	4
4	1	2	2	2
5	8	8	9	9
6	5	12	8	10
8	9	6	12	6
9	10	5	11	12
10	14	9	13	8
11	6	10	6	11
12	12	13	5	5
13	13	11	14	14
14	15	14	10	13
15	11	16	21	16
16	16	17	16	15
17	17	21	15	17
20	20	29	26	21
21	22	23	17	26
22	21	26	29	23
23	26	15	23	29
24	23	20	31	20
25	38	24	20	24
26	31	22	38	38
29	24	31	22	22
31	32	25	35	31
32	29	34	24	35
33	35	36	32	32
34	34	32	25	25
35	25	33	34	33
36	33	38	36	34
38	36	35	33	36

Table L.2(f) – Reliability of best module ranges of LDST for a population size of 200.

Expert	OX	OX2	PMX	OX’
Reliability	96.08%	95.64%	92.34%	**96.97%**
1	1	1	1	1
2	2	2	4	2
4	9	4	8	4
5	4	5	2	5
6	6	8	5	9
8	8	9	9	6
9	10	6	12	10
10	11	10	6	8
11	12	14	10	14
12	14	12	13	12
13	13	13	17	13
14	17	16	14	16
15	5	20	16	11
16	16	17	11	17
17	21	11	15	15
20	22	15	29	23
21	15	21	23	20
22	23	23	22	26
23	20	24	26	24
24	26	29	20	21
25	24	22	38	22
26	29	35	24	25
29	25	25	21	33
31	34	38	31	29
32	31	26	35	38
33	32	31	32	31
34	35	32	25	32
35	33	33	34	34
36	38	34	36	35
38	36	36	33	36

M. Reliable Module Ranges with Reliability Percentages of the Control Data

Table M.1(a) – Reliability of best module ranges of SDST for a population size of 100.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	95.83	96.81	95.10	**97.79**
1	1	1	1	1
3	3	5	4	3
4	4	4	3	4
5	5	9	5	5
7	11	3	11	7
9	7	7	7	9
11	12	11	12	11
12	9	12	9	12
14	14	14	14	14
19	19	19	19	29
21	29	29	30	19
22	22	21	29	21
29	34	22	22	22
30	30	30	21	34
34	21	34	40	36
36	36	40	36	30
40	40	36	34	40

Table M.1 (b) – Reliability of best module ranges of SDST for a population size of 120.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	96.56	**98.04**	94.36	96.81
1	1	1	1	1
3	4	4	4	4
4	3	5	3	3
5	11	3	5	5
7	5	7	11	14
9	7	11	7	7
11	12	12	14	11
12	14	9	12	9
14	9	14	9	12
19	19	19	19	19
21	22	22	29	21
22	21	21	21	22
29	29	29	34	29
30	30	34	36	30
34	36	30	22	36
36	34	36	30	34
40	40	40	40	40

Table M.1 (c) – Reliability of best module ranges of SDST for a population size of 140.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	95.83	97.30	96.08	**98.28**
1	1	1	1	1
3	4	4	4	3
4	3	5	3	4
5	7	3	5	5
7	5	11	7	7
9	11	7	11	11
11	12	9	9	12
12	9	12	12	9
14	29	14	14	14
19	14	19	19	22
21	19	21	21	19
22	21	29	29	21
29	30	22	34	29
30	22	30	36	30
34	34	40	22	36
36	36	36	40	34
40	40	34	30	40

Table M.1 (d) – Reliability of best module ranges of SDST for a population size of 160.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	97.06	**97.30**	95.59	96.81
1	1	1	1	1
3	4	4	4	4
4	5	3	3	5
5	3	5	14	3
7	7	11	5	7
9	9	7	7	11
11	11	12	9	9
12	12	9	11	12
14	14	14	12	14
19	19	19	19	19
21	21	21	22	22
22	29	29	21	29
29	34	22	29	30
30	22	30	34	21
34	40	40	30	36
36	30	36	36	40
40	36	34	40	34

Table M.1 (e) – Reliability of best module ranges of SDST for a population size of 180.

Expert	OX	OX2	PMX	OX’
Reliability	96.08	**98.53**	94.36	97.80
1	1	1	1	1
3	4	4	3	3
4	3	3	4	4
5	9	5	5	5
7	11	7	11	11
9	5	9	12	7
11	7	11	7	9
12	12	12	14	12
14	14	14	9	14
19	19	19	19	19
21	22	22	29	22
22	30	21	21	21
29	29	29	22	29
30	21	30	36	34
34	34	40	40	40
36	36	36	34	30
40	40	34	30	36

Table M.1(f) – Reliability of best module ranges of SDST for a population size of 200.

Expert	OX	OX2	PMX	OX’
Reliability	95.34	**96.81**	96.07	95.34
1	1	1	1	1
3	3	4	4	3
4	4	3	3	4
5	5	5	5	5
7	11	7	11	11
9	14	11	7	7
11	7	12	12	12
12	9	14	9	14
14	12	9	14	22
19	19	19	19	9
21	22	29	29	19
22	21	21	21	21
29	36	22	22	29
30	30	36	40	34
34	29	30	30	30
36	34	34	36	40
40	40	40	34	36

Table M.2(a) – Reliability of best module ranges of LDST for a population size of 100.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	93.94	**97.14**	92.70	96.57
1	1	1	11	1
3	4	4	1	3
4	5	3	3	4
5	3	5	4	5
7	7	11	7	7
9	14	7	5	11
11	11	14	13	14
13	18	13	23	16
14	23	9	17	9
16	25	16	25	22
17	13	19	14	19
18	9	18	18	21
19	22	23	16	17
21	19	21	9	18
22	17	25	29	23
23	28	22	19	25
25	16	28	28	13
26	21	17	21	26
28	26	31	22	29
29	29	29	33	30
30	30	33	26	28
31	31	26	30	31
33	33	30	31	33
34	34	34	34	34
35	35	35	35	37
37	37	37	37	38
38	38	38	38	35
39	39	39	39	39
40	41	40	40	40
41	40	41	41	41

Table M.2 (b) – Reliability of best module ranges of LDST for a population size of 120.

Expert	**OX**	**OX2**	**PMX**	**OX'**
Reliability	92.74	**96.75**	92.65	95.81
1	1	1	1	1
3	4	4	4	4
4	3	14	14	7
5	5	3	7	5
7	11	11	5	3
9	7	7	16	14
11	9	5	9	11
13	14	13	13	19
14	23	22	17	9
16	17	9	22	22
17	16	16	3	16
18	13	18	18	23
19	30	19	19	13
21	31	23	25	17
22	18	21	26	25
23	19	17	23	18
25	25	25	21	26
26	26	26	11	21
28	28	30	30	30
29	29	29	28	31
30	21	28	31	29
31	22	31	33	28
33	33	33	29	33
34	34	34	34	34
35	35	35	35	35
37	37	37	37	37
38	40	38	38	38
39	38	39	39	39
40	39	40	40	40
41	41	41	41	41

Table M.2(c) – Reliability of best module ranges of LDST for a population size of 140.

Expert	OX	OX2	PMX	OX’
Reliability	93.10	**96.93**	93.77	96.08
1	1	1	1	1
3	3	4	4	4
4	4	3	5	3
5	14	14	3	11
7	21	11	7	5
9	5	5	18	7
11	11	7	17	14
13	7	9	11	9
14	9	13	13	16
16	16	16	14	13
17	23	17	9	19
18	22	23	16	18
19	13	19	19	23
21	17	25	25	17
22	18	22	26	33
23	31	28	33	25
25	29	18	23	22
26	19	21	28	21
28	25	30	21	31
29	26	29	34	29
30	30	26	30	30
31	34	34	31	26
33	28	31	22	28
34	33	33	29	34
35	35	35	35	37
37	39	37	37	35
38	37	38	38	38
39	38	39	39	39
40	40	40	40	40
41	41	41	41	41

Table M.2 (d) – Reliability of best module ranges of LDST for a population size of 160.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	93.86	**96.00**	95.90	95.46
1	1	1	1	1
3	4	4	4	4
4	3	14	5	14
5	5	3	3	3
7	14	5	7	5
9	7	7	11	7
11	11	11	13	9
13	17	9	9	11
14	9	13	14	16
16	16	23	19	13
17	13	18	17	19
18	23	22	22	23
19	30	19	23	22
21	31	16	25	25
22	18	17	16	21
23	19	21	28	30
25	25	25	29	18
26	21	26	21	26
28	26	30	18	17
29	28	29	31	29
30	29	33	26	28
31	22	31	34	33
33	33	34	33	31
34	34	37	30	34
35	35	28	35	35
37	37	35	37	37
38	38	38	38	38
39	39	39	40	39
40	40	40	39	40
41	41	41	41	41

Table M.2 (e) – Reliability of best module ranges of LDST for a population size of 180.

Expert	**OX**	**OX2**	**PMX**	**OX’**
Reliability	92.83	**97.64**	93.94	96.35
1	1	1	1	1
3	4	4	4	4
4	3	5	3	14
5	14	3	11	5
7	7	7	5	3
9	5	17	7	7
11	11	11	14	11
13	16	13	19	16
14	9	14	16	9
16	17	9	13	13
17	23	16	9	18
18	22	19	17	23
19	13	23	23	25
21	30	22	22	19
22	31	21	28	22
23	18	18	18	17
25	19	25	30	21
26	25	28	25	29
28	26	31	29	31
29	28	30	31	26
30	29	26	21	30
31	33	33	37	28
33	21	29	26	33
34	34	34	33	34
35	35	35	41	37
37	37	37	34	35
38	38	38	39	39
39	40	39	35	38
40	39	40	38	41
41	41	41	40	40

Table M.2(f) – Reliability of best module ranges of LDST for a population size of 200.

Expert	**OX**	**OX2**	**PMX**	**OX'**
Reliability	93.37	95.99	93.77	**97.41**
1	1	1	1	1
3	4	4	4	4
4	5	5	14	3
5	3	3	3	5
7	7	11	5	7
9	11	7	11	9
11	14	9	9	17
13	19	13	7	11
14	9	14	18	13
16	16	16	25	14
17	13	19	16	16
18	31	23	19	23
19	23	21	13	18
21	22	30	17	25
22	21	22	23	19
23	30	25	28	21
25	17	17	30	22
26	18	18	21	31
28	25	28	33	30
29	26	33	26	29
30	28	26	29	33
31	29	29	22	26
33	33	31	31	28
34	34	34	34	34
35	40	37	35	35
37	35	35	41	37
38	37	38	37	38
39	38	39	38	39
40	39	40	39	40
41	41	41	40	41

N. Comparison Graphics of Training and Control Data for SDST

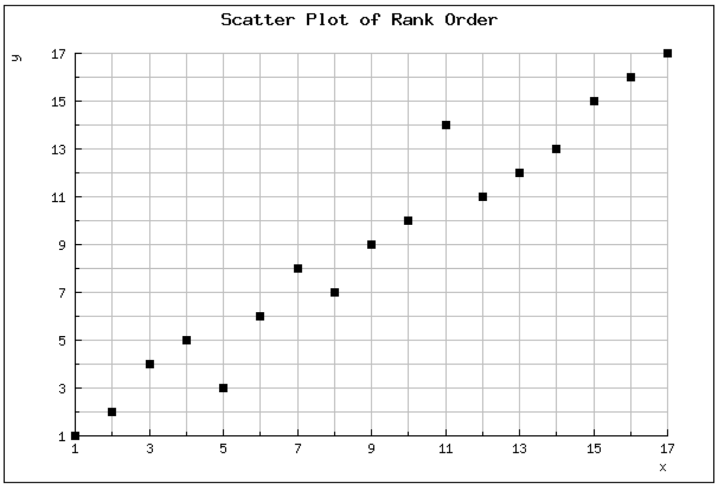


Figure N.1(a) –Comparing the most reliable module range result of training data – SDST for a population size of 100 with Expert's Suggestion. The solution is 97.55% reliable in OX'.

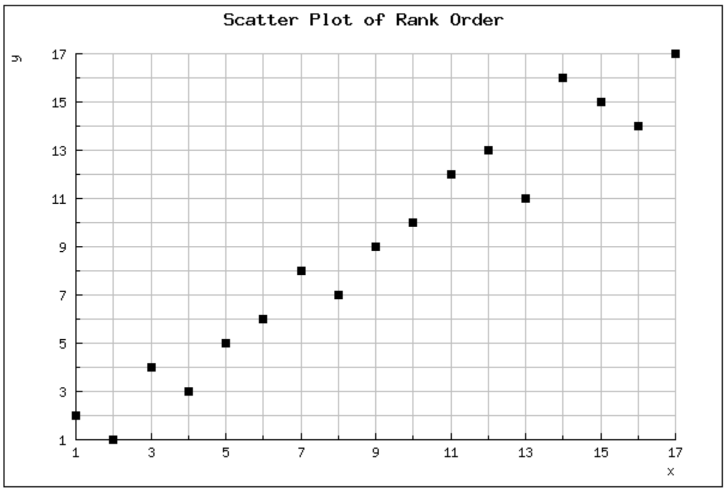


Figure N.1(b) –Comparing the most reliable module range result of training data – SDST for a population size of 120 with Expert's Suggestion. The solution is 97.55% reliable in OX2.

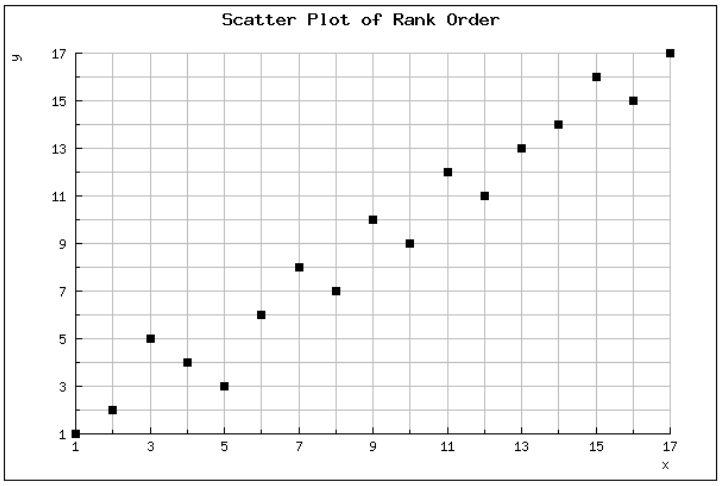


Figure N.1(c) –Comparing the most reliable module range result of training data – SDST for a population size of 140 with Expert's Suggestion. The solution is 98.04% reliable in OX'.

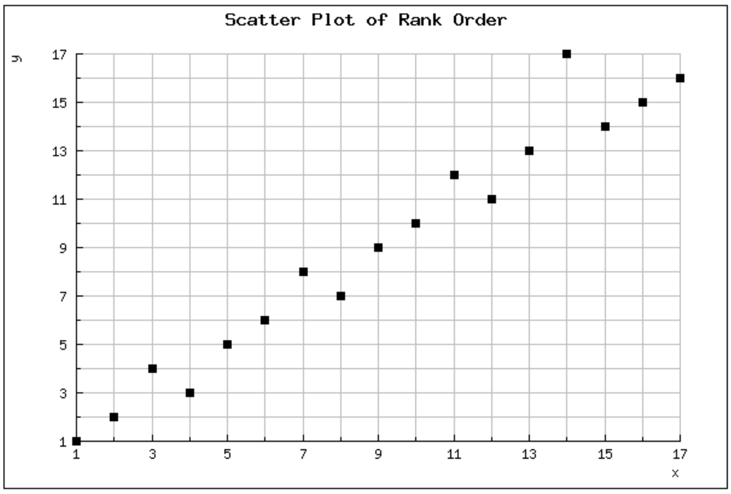


Figure N.1(d) –Comparing the most reliable module range result of training data – SDST for a population size of 160 with Expert's Suggestion. The solution is 97.79% reliable in OX2.

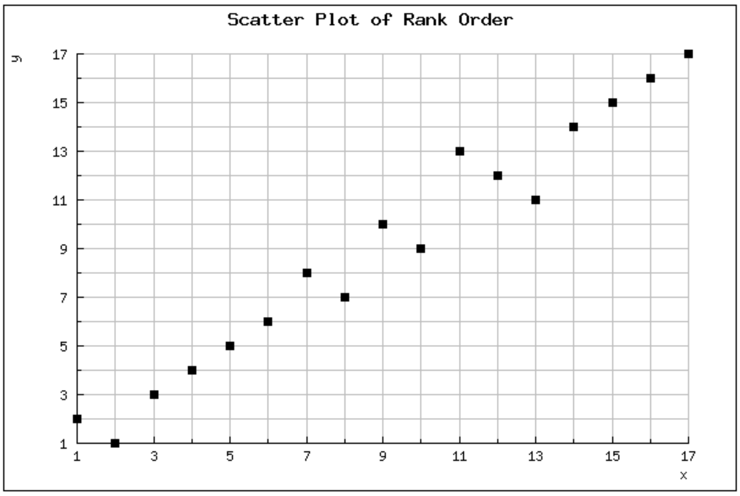


Figure N.1(e) –Comparing the most reliable module range of training data – SDST result for a population size of 200 with Expert’s Suggestion. The solution is 98.28% reliable in OX2.

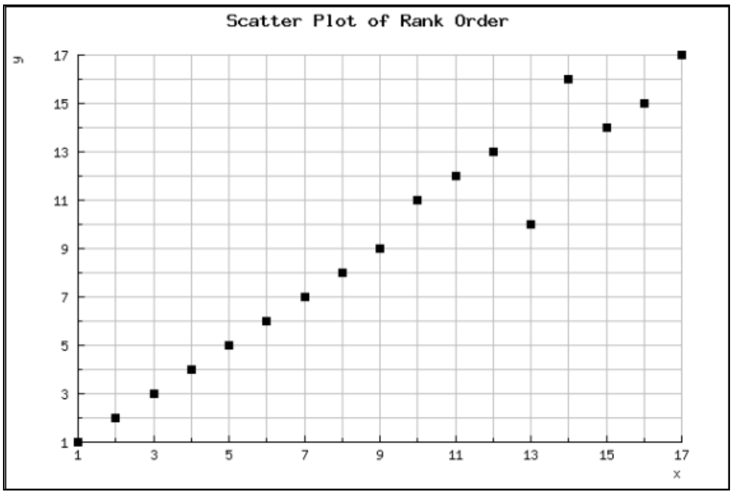


Figure N.2(a) –Comparing the most reliable module range of control data – SDST result for a population size of 100 with Expert’s Suggestion. The solution is 97.79% reliable in OX’.

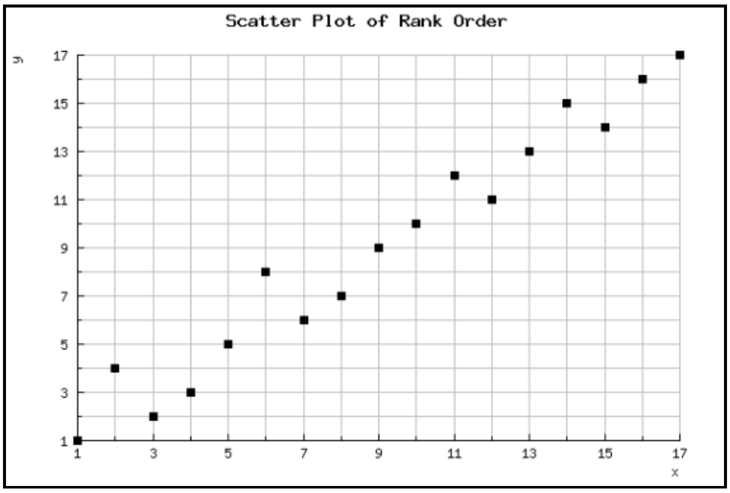


Figure N.2(b) –Comparing the most reliable module range of control data – SDST result for a population size of 120 with Expert’s Suggestion. The solution is 98.04% reliable in OX2.

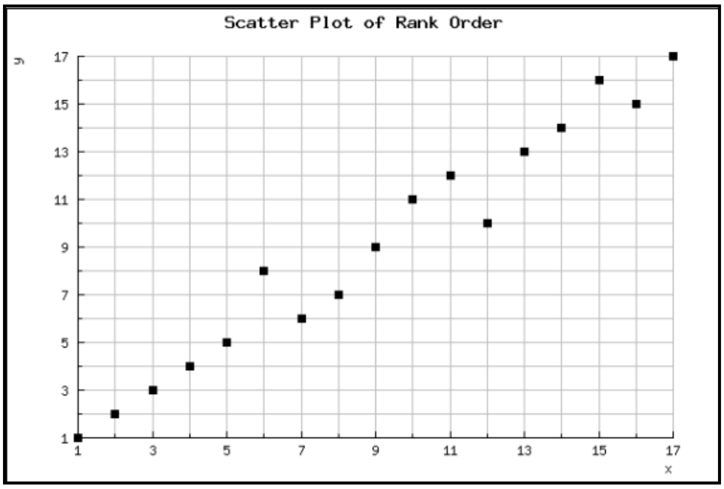


Figure N.2(c) –Comparing the most reliable module range of control data – SDST result for a population size of 140 with Expert’s Suggestion. The solution is 98.28% reliable in OX’.

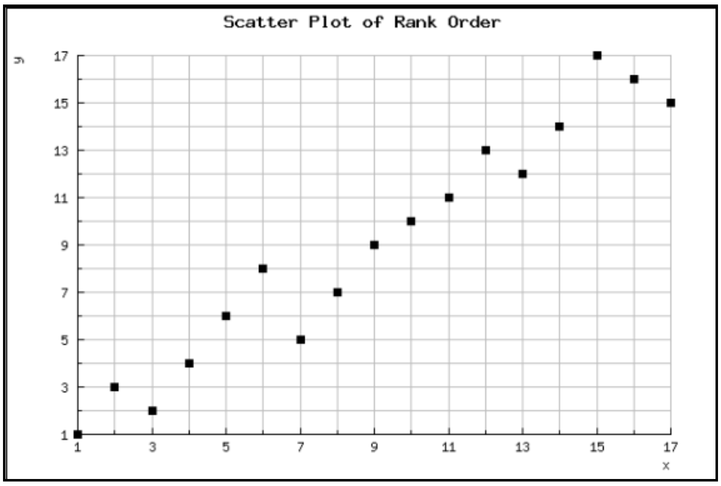


Figure N.2(d) –Comparing the most reliable module range of control data – SDST result for a population size of 160 with Expert’s Suggestion. The solution is 97.30% reliable in OX2.

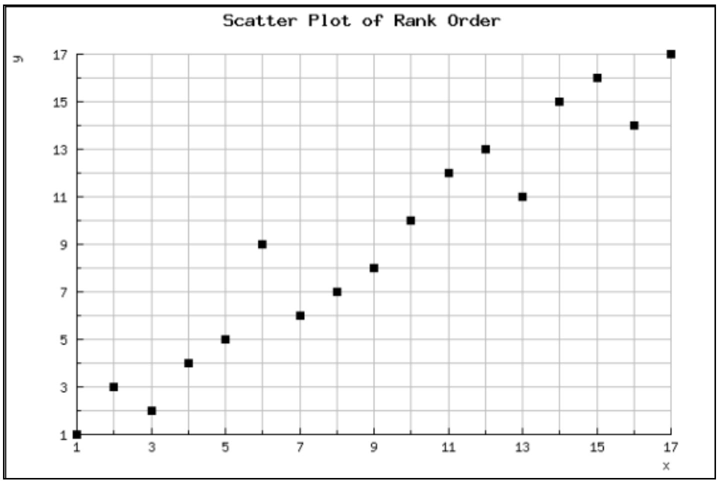


Figure N.2(e) –Comparing the most reliable module range of control data – SDST result for a population size of 200 with Expert’s Suggestion. The solution is 96.81% reliable in OX2.

O. Comparison Graphics of Training and Control Data for LDST

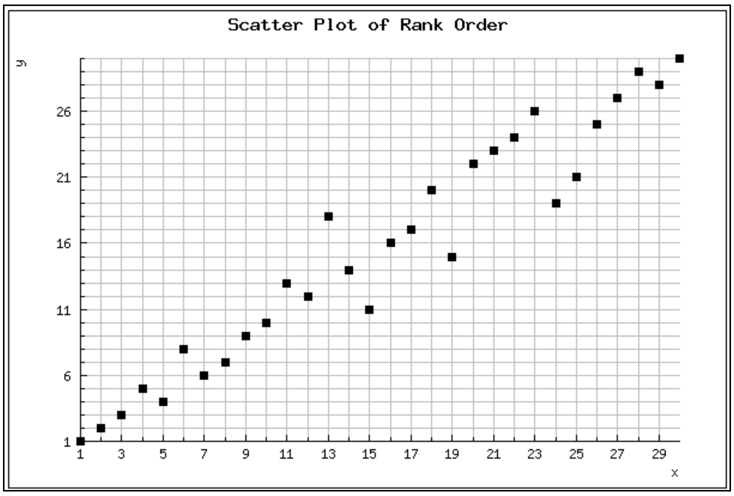


Figure O.1(a) –Comparing the most reliable module range result of training data – LDST for a population size of 100 with Expert's Suggestion. The solution is 96.93% reliable in OX2.

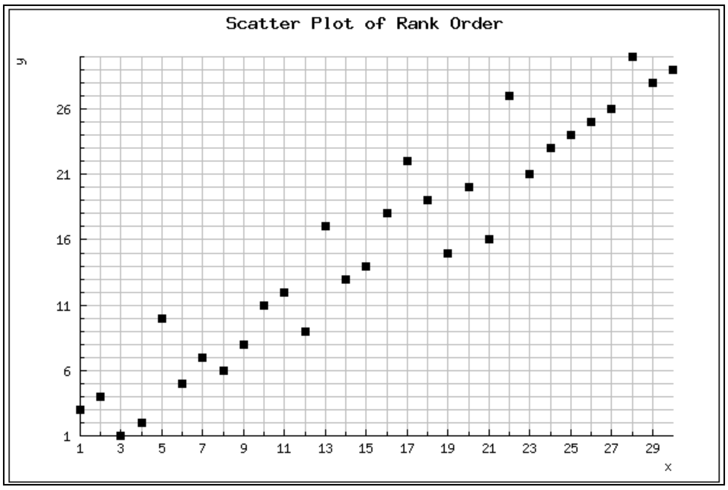


Figure O.1(b) –Comparing the most reliable module range result of training data – LDST for a population size of 120 with Expert's Suggestion. The solution is 95.86% reliable in OX2.

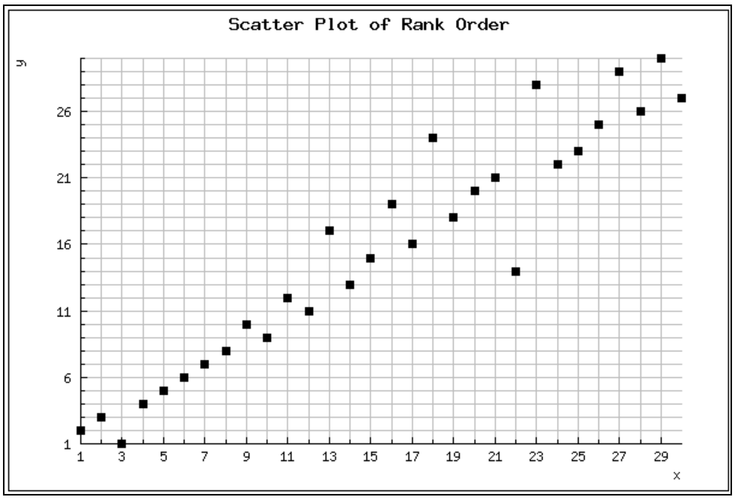


Figure O.1(c) –Comparing the most reliable module range result of training data – LDST for a population size of 160 with Expert's Suggestion. The solution is 95.77% reliable in OX2.

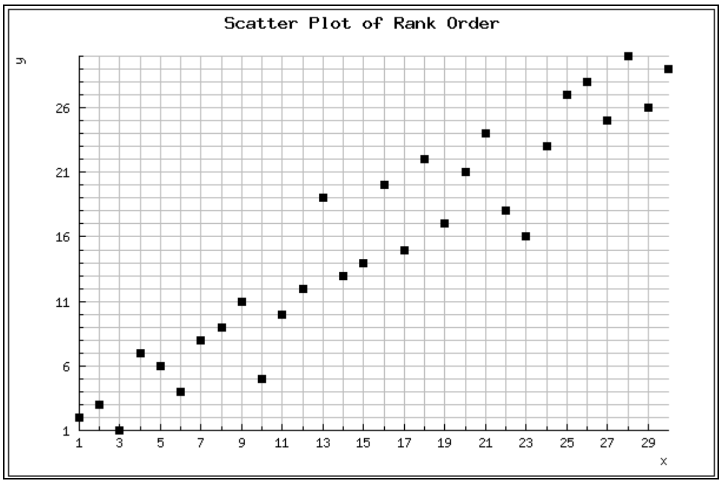


Figure O.1(d) –Comparing the most reliable module range result of training data – LDST for a population size of 180 with Expert's Suggestion. The solution is 94.84% reliable in OX2.

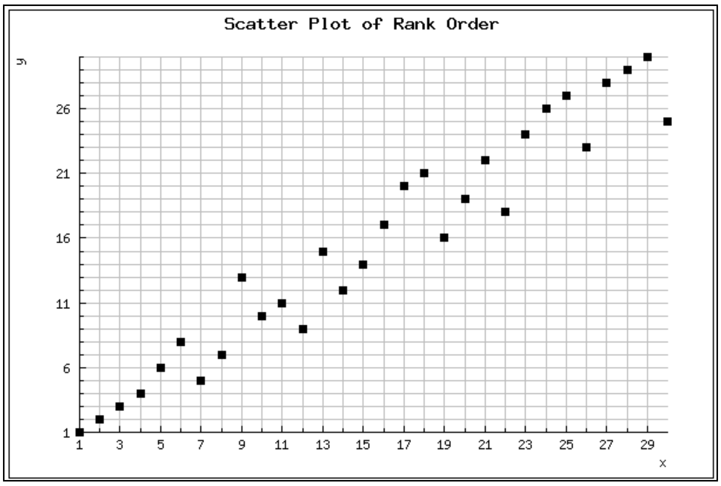


Figure O.1(e) –Comparing the most reliable module range result of training data – LDST for a population size of 200 with Expert's Suggestion. The solution is 96.97% reliable in OX'.

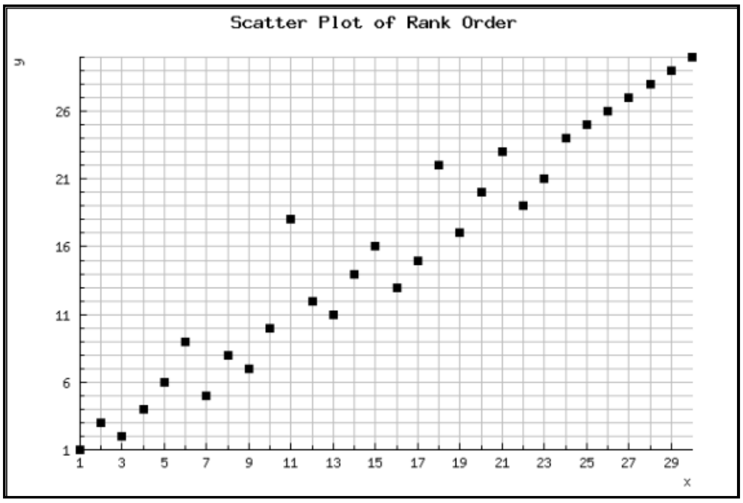


Figure O.2(a) –Comparing the most reliable module range result of control data – LDST for a population size of 100 with Expert's Suggestion. The solution is 97.14% reliable in OX2.

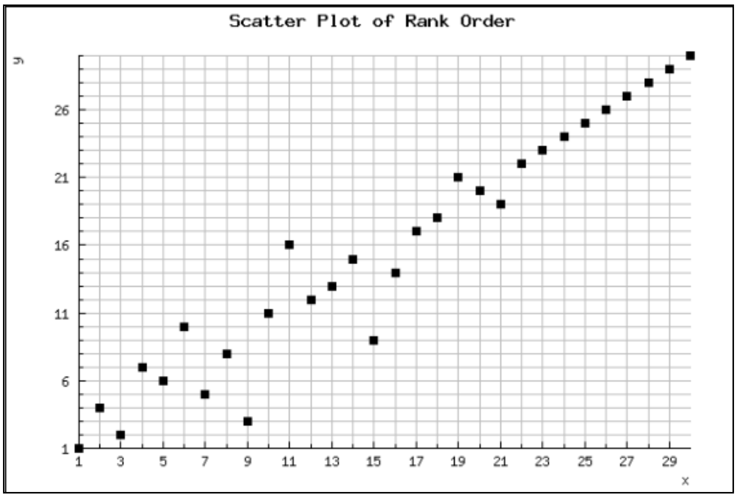


Figure O.2(b) –Comparing the most reliable module range result of control data – LDST for a population size of 120 with Expert's Suggestion. The solution is 96.75% reliable in OX2.

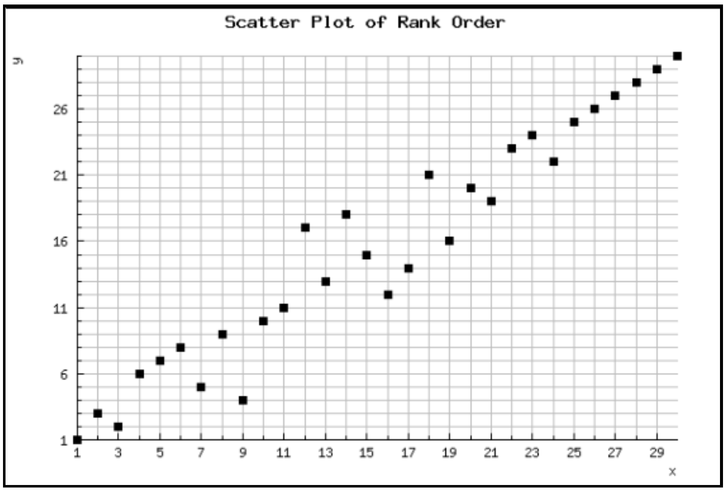


Figure O.2(c) –Comparing the most reliable module range result of control data – LDST for a population size of 140 with Expert's Suggestion. The solution is 96.93% reliable in OX2.

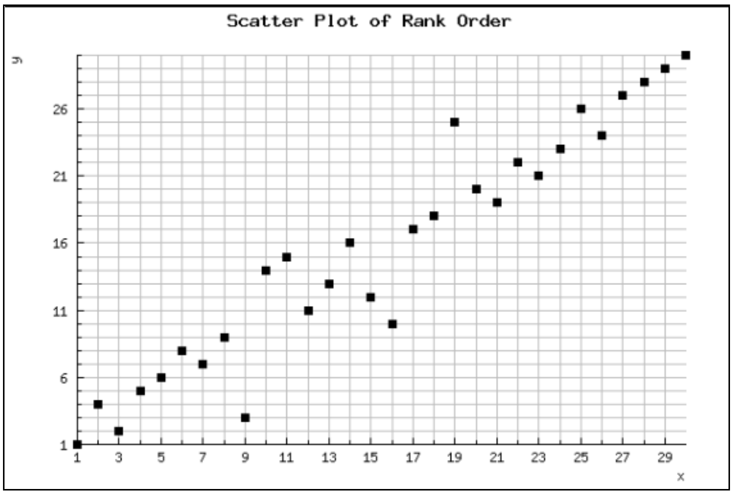


Figure O.2(d) –Comparing the most reliable module range result of control data – LDST for a population size of 160 with Expert's Suggestion. The solution is 96.00% reliable in OX2.

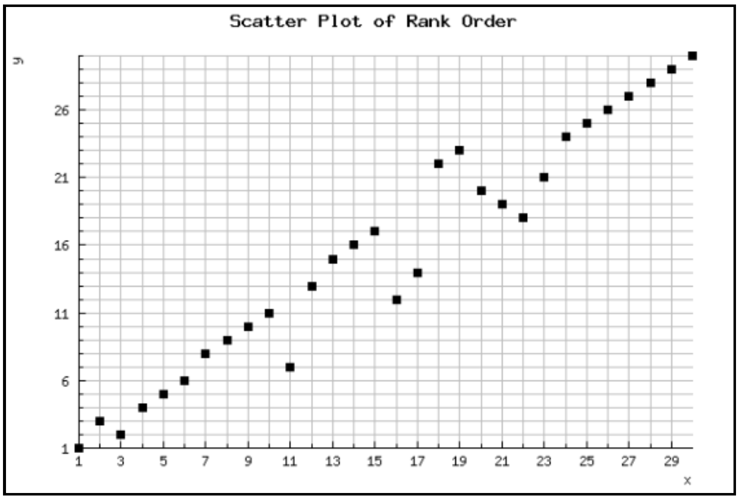


Figure O.2(e) –Comparing the most reliable module range result of control data – LDST for a population size of 200 with Expert's Suggestion. The solution is 97.41% reliable in OX'.

P. RUNTIME GRAPHICS OF TRAINING AND CONTROL DATA

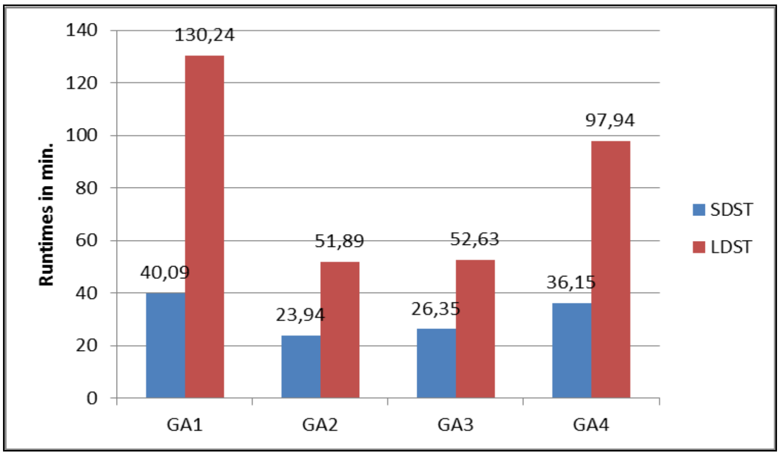


Figure P.1(a) – Runtime graphs showing the dataset growth for training data of 120 individuals.

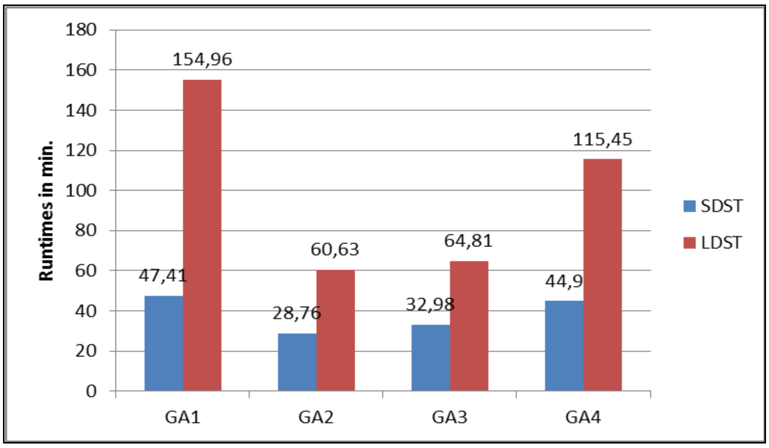


Figure P.1(b) – Runtime graphs showing the dataset growth for training data of 140 individuals.

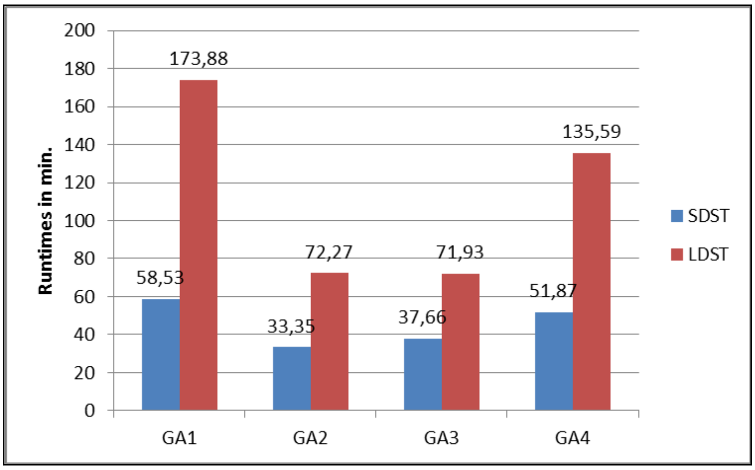


Figure P.1(c)– Runtime graphs showing the dataset growth for training data of 160 individuals.

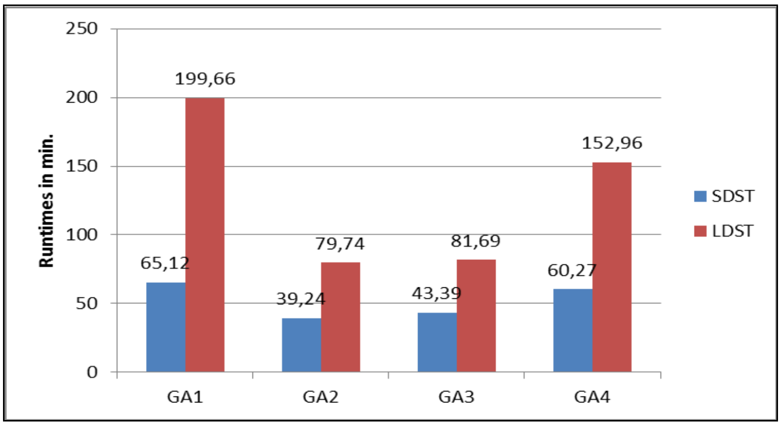


Figure P.1(d) – Runtime graphs showing the dataset growth for training data of 180 individuals.

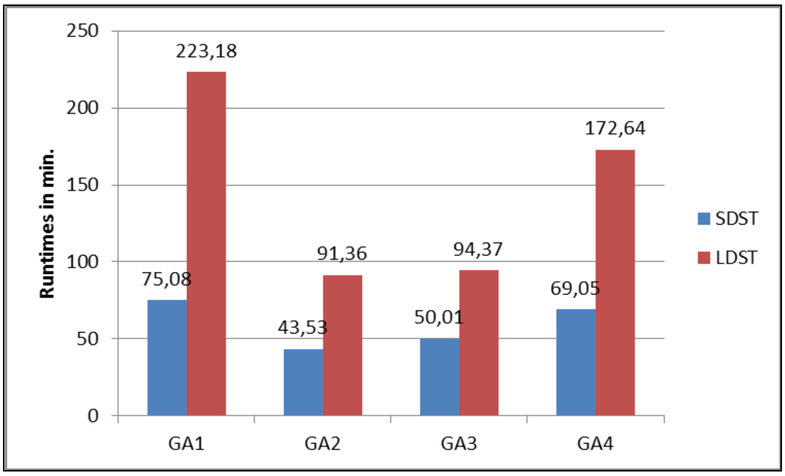


Figure P.1(e) – Runtime graphs showing the dataset growth for training data of 200 individuals.

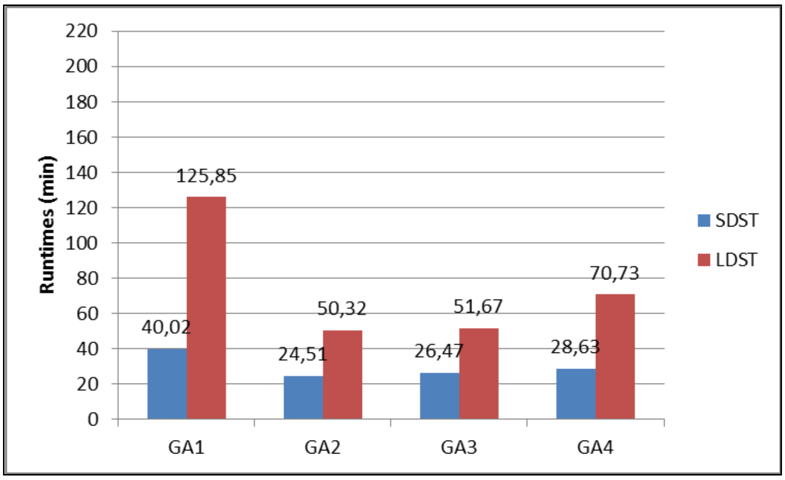


Figure P.2(a) – Runtime graphs showing the dataset growth for control data of 120 individuals.

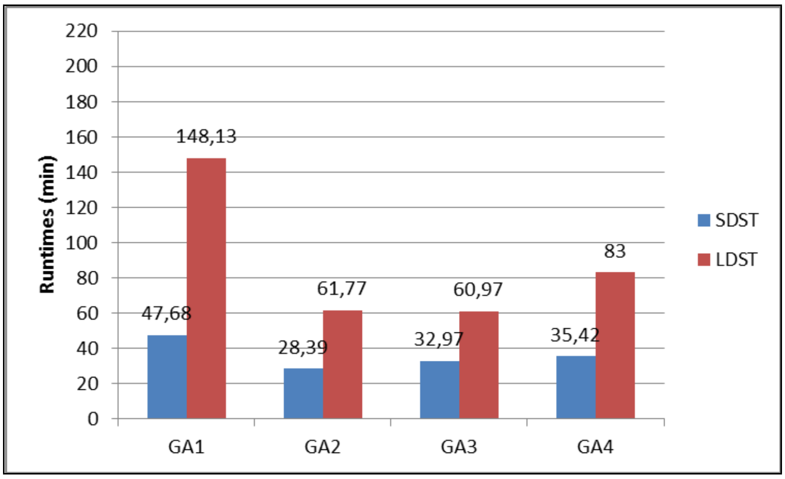


Figure P.2(b) – Runtime graphs showing the dataset growth for control data of 140 individuals.

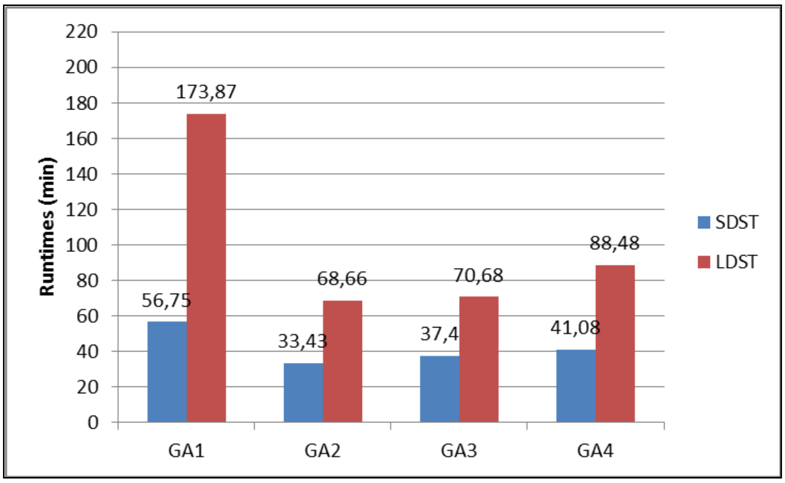


Figure P.2(c) – Runtime graphs showing the dataset growth for control data of 160 individuals.

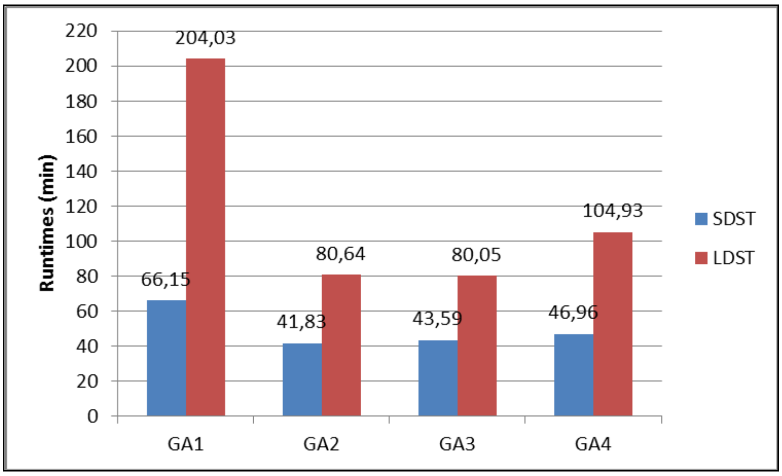


Figure P.2(d) – Runtime graphs showing the dataset growth for control data of 180 individuals.

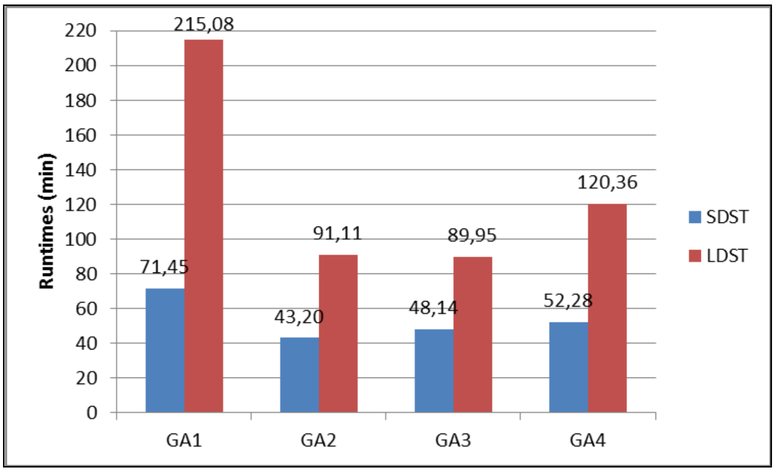


Figure P.2(e) – Runtime graphs showing the dataset growth for control data of 200 individuals.

Q. Trends of Increase in Runtimes for the Training and Control Data

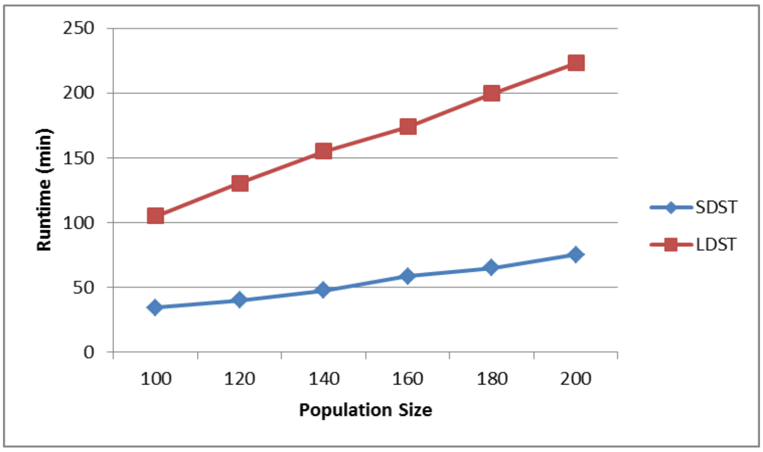


Figure Q.1(a) – Trend of increase in runtime values of training data for OX.

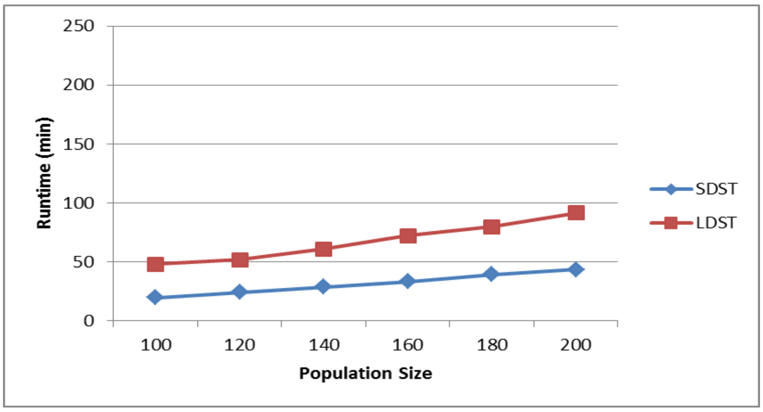


Figure Q.1(b) – Trend of increase in runtime values of training data for OX2.

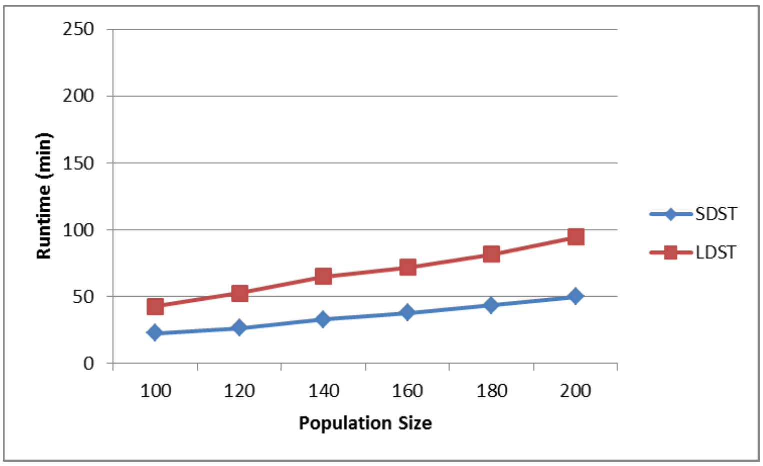


Figure Q.1(c) – Trend of increase in runtime values of training data for PMX.

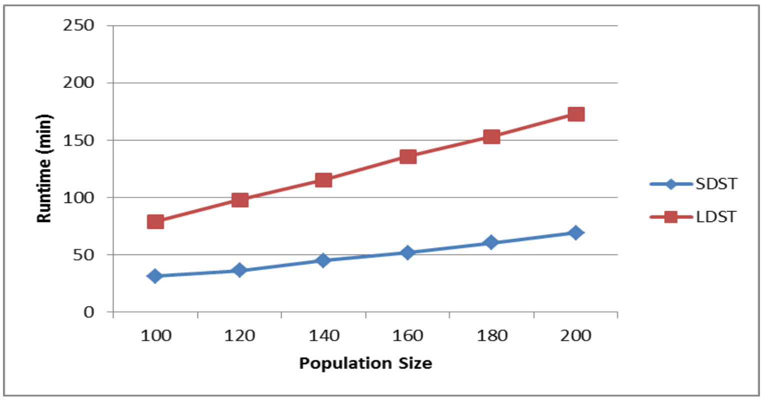


Figure Q.1(d) – Trend of increase in runtime values of training data for OX’.

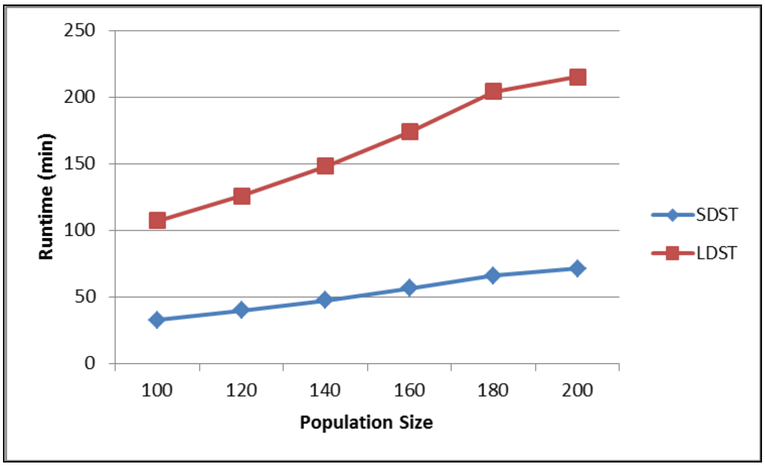


Figure Q.2(a) – Trend of increase in runtime values of control data for OX.

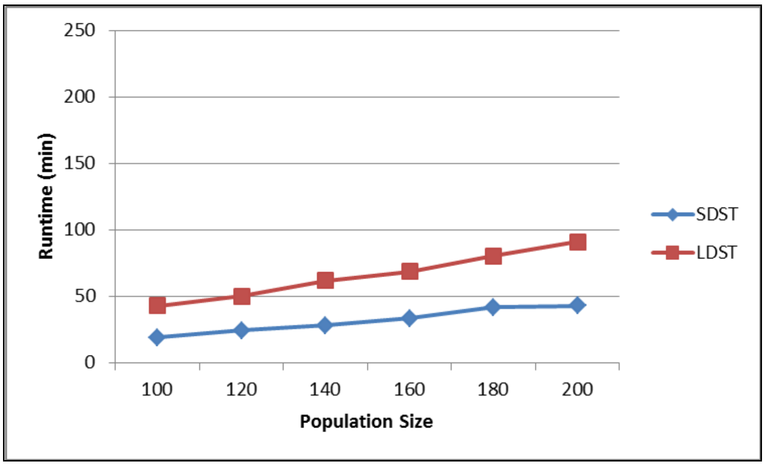


Figure Q.2(b) – Trend of increase in runtime values of control data for OX2.

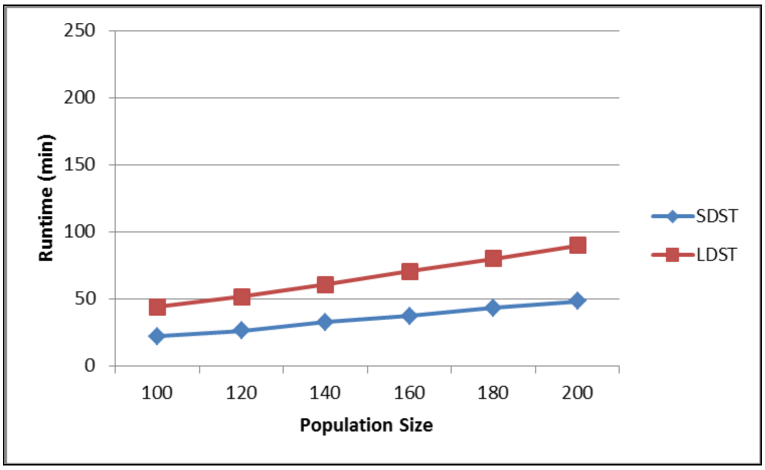


Figure Q.2(c) – Trend of increase in runtime values of control data for PMX.

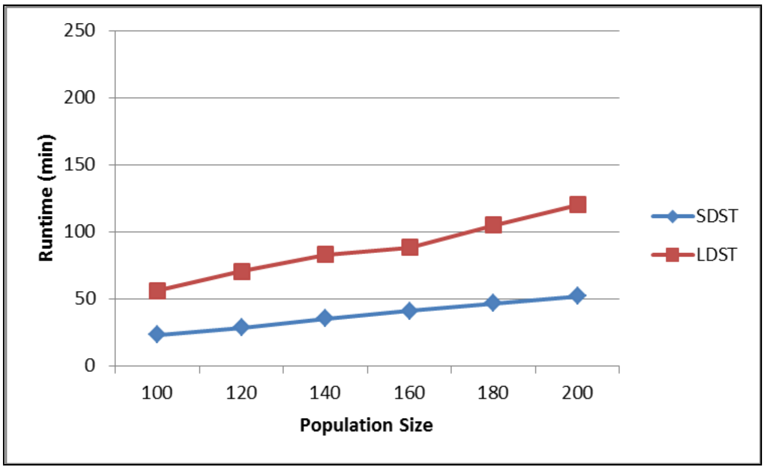


Figure Q.2(d) – Trend of increase in runtime values of control data for OX'.