UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

SYNTHESIS OF PLANAR MECHANISMS USING EVOLUTIONARY ALGORITHMS

M. Sc. THESIS
IN
MECHANICAL ENGINEERING

BY
HÜSEYİN ERDOĞAN
SEPTEMBER 2011

Synthesis of Planar Mechanisms using Evolutionary Algorithms

M.Sc. Thesis
In
Mechanical Engineering
University of Gaziantep

Supervisor
Prof. Dr. Lale Canan DÜLGER

By

Hüseyin ERDOĞAN

September 2011

T.C.
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES
(Mechanical Engineering)

Name of the thesis : Synthesis of Planar Mechanisms using Evolutionary Algorithms
Name of the student : Hüseyin ERDOĞAN
Exam date : 09.09.2011

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Ramazan KOÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. L. Canan DÜLGER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Lale Canan DÜLGER
Supervisor

Examining Committee Members	Signature
Prof.Dr.İ.Halil GÜZELBEY	
Prof.Dr.L.Canan DÜLGER	
Assoc.Prof.Dr.Cem GÜNEŞOĞLU	
Asist.Prof.Dr.Ahmet ERKLİĞ	
Assist.Prof.Dr.A.Tolga BOZDANA	

ABSTRACT

SYNTHESIS OF PLANAR MECHANISMS USING EVOLUTIONARY ALGORITHMS

ERDOĞAN Hüseyin

M Sc. in Mechanical Engineering

Supervisor: Prof. Dr. L. Canan DÜLGER

September 2011, 60 pages

There are many optimization methods, but use of Evolutionary Algorithms (EAs) has been increased due to the best results they offer in recent years. They have been very popular for solving nonlinear problems in engineering. EAs show varying degrees of success for different engineering applications. The optimal solution depends on the formulation of the problem. Genetic algorithm (GA) is an evolutionary optimization technique based on genetics principles and natural selection. Evolutionary algorithms are studied in general. Genetic Algorithm Toolbox, Optimization Toolbox Matlab© is explored and presented with examples in this study.

This study presents an optimization approach for synthesis of planar mechanisms. Since four bar mechanism is simple and practically important mechanism, it is chosen in here. This mechanism is studied with the constraints assigned. Genetic Algorithm (GA) is applied during optimization study. GA is compared with nonlinear constrained numerical optimization command; *fmincon* in Matlab©. Three different case studies are presented. Different target points are considered for mechanism operation. These mechanisms are drawn using Excel© spread sheet to see their animations also. Performances of both algorithms are compared for their general use in similar problems based on mechanism synthesis.

Key Words: Evolutionary Algorithms (EAs), Synthesis of planar mechanisms, Genetic Algorithm, Engineering Optimization, Four bar mechanism

ÖZET

SEZGİSEL ALGORİTMALARLA DÜZLEMSEL MEKANİZMALARIN SENTEZİ

ERDOĞAN Hüseyin
Yüksek Lisans Tezi, Makine Müh. Bölümü
Tez Yöneticisi: Prof. Dr. L. Canan DÜLGER
Eylül 2011, 60 sayfa

Bir çok en iyileştirme yöntemi bulunmaktadır. Ancak sundukları en iyi çözümlerden dolayı sezgisel algoritmaların kullanımı son yıllarda artmıştır. Özellikle mühendislikte doğrusal olmayan problemlerin çözümünde kullanılmaları yaygındır. Sezgisel algoritmalar farklı mühendislik problemlerinin çözümünde farklı başarı dereceleri ile karşımıza çıkmaktadır. En iyi çözüm problemin tanımına bağlı olarak değişebilir. Genetik algoritma sezgisel bir algoritma olup, genetik prensiplere ve doğal seçime dayanan bir tekniktir. Çalışmada genel olarak GA ile Matlab tabanlı Optimizasyon araç kutusu doğrusal olmayan düzlemsel mekanizma sentezinde kullanılmış ve karşılaştırılmıştır.

Çalışma düzlemsel mekanizmaların sentezinde en iyileştirme yaklaşımını sunmaktadır. Dört çubuk mekanizması basit ve pratikte önemli bir mekanizma olduğundan dolayı seçilmiştir. Mekanizma kısıtlarla alınarak genetik algoritma (GA) ve Matlab Optimizasyon araç kutusu ile en uygun mekanizma boyutları bulunmuştur. Kısıtlı doğrusal olmayan kısıt sayısal en iyileştirmede GA ve Matlab komutu *fmincon* birlikte çalışılmıştır. Bu mekanizmalar Excel programı kullanılarak çizil ve GA ile Matlab sentez tabanlı uygulamalarda kullanımı açısından karşılaştırılmıştır.

Anahtar Kelimeler: Sezgisel algoritmalar, Düzlemsel mekanizmaların sentezi, Genetik algoritma, Mühendislikte en iyileştirme, Dört çubuk mekanizması

ACKNOWLEDGMENTS

I am thankful to my supervisor Prof.Dr. Lale Canan DÜLGER who has provided continuous support and encouragement throughout my study. She has always helped me tackling the problems during the thesis. It has been hard time for me to complete the study. I have never lost my ambition during the study.

My special thanks to my family and my children.

CONTENTS Page

LIST OF TABLES **Page**

LIST OF FIGURES **Page**

CHAPTER 1
INTRODUCTION

1.1. Introduction

The purpose of this study is to perform synthesis of mechanical linkages using Evolutionary Algorithms. Some recent studies on the subject covering more than ten years is surveyed. The optimum synthesis of a mechanism requires a repeated analysis to find the best possible one to meet requirements. A simulation study will be performed on a four bar linkage. The linkage parameters will be tabulated as a guide for the user. The computational synthesis methods are also applied [1].

The science of motion is related with the analysis and synthesis of mechanisms in study of Kinematics. It also deals with the relative geometric displacements of points and links of a mechanism. There is a need for an approach to synthesis by using the direct and certain methods. In some special cases of the mechanisms the analytical methods are used that these are serial articulated joints or certain parallel mechanisms. Dimensional Synthesis looks for determining optimal dimensions of a prescribed type of mechanism. The type and dimensional levels are the main factors in the mechanisms for the study of kinematic synthesis of mechanisms [2].

The ready web browser like a graphical user interface(GUI) for system definition and presentation of the analysis results can be used. This study shows some of the results like textual, pictorial, graphical and animation information. In forward steps, our study may contain a number of modules in system simulation and planar mechanism analysis. Two main components are included for mechanism studies; the design or synthesis of the mechanism which is based on requirements or specification and the analysis of the designed or synthesized mechanism.

Here the four-bar linkage will be studied for the most common planar mechanism. When the mechanism synthesis is studied, the mechanism parameters are defined like that the coupler point of the mechanism moves through a specified number and coordinates of precision points on the basis of given certain inputs. The synthesis and analysis on planar mechanism is performed according to the number of precision positions that the mechanism is working at this interval positions. The developed units can synthesize and analyze planar four bar linkages by using the motion generation methodology subject to the following number of defined positions. These are two precision positions, three precision positions with fixed pivots, three precision positions without fixed pivots, four precision positions and five precision positions. To solve a kinematic synthesis problem for a clearly defined design task that the simulation and evaluation is a prerequisite for brasing different mechanism topologies and parameters. To see whether the mechanism is suitable for task it is necessary to use the simulation directly [2].

1.2. Synthesis of Planar Mechanisms

A commonly used mechanisms in a number of machinery is four bar mechanism whic has four revolute joints. It can be seen with numerous machinery applications. There is a relationship of the angular rotations of the links that is connected to the fixed link (*correlation of crank angles or function generation)*. If there is not any connection to the fixed link, it is called the *coupler link.* This position of the coupler link can be used as the output of the four bar mechanism. The link lengths determine the motion characteristics of a four bar mechanism according to the Grashof's theorem. The link lengths are the function of the type of motion and are identified for a four bar chain as follows [3]. Here l is the longest link length, s is the shortest link length, p and q are the two intermediate link lengths. The input-output equation of a four bar is taken as by looking at link legths. Figure 1.1 shows all possible mechanisms with given configuration.

(i) <u>*Case 1: l+s<p+q*</u>

The longest and shortest link lengths are summed, and this is less than the sum of the other two intermediate link lengths[1]. Two different crank-rocker mechanisms as possible. The shortest link which is the crank, the other fixed link is the adjacent link. One double-crank (drug-link) and the other shortest link which is called as the frame. One double-rocker mechanism; the frame is the opposite and the shortest link.

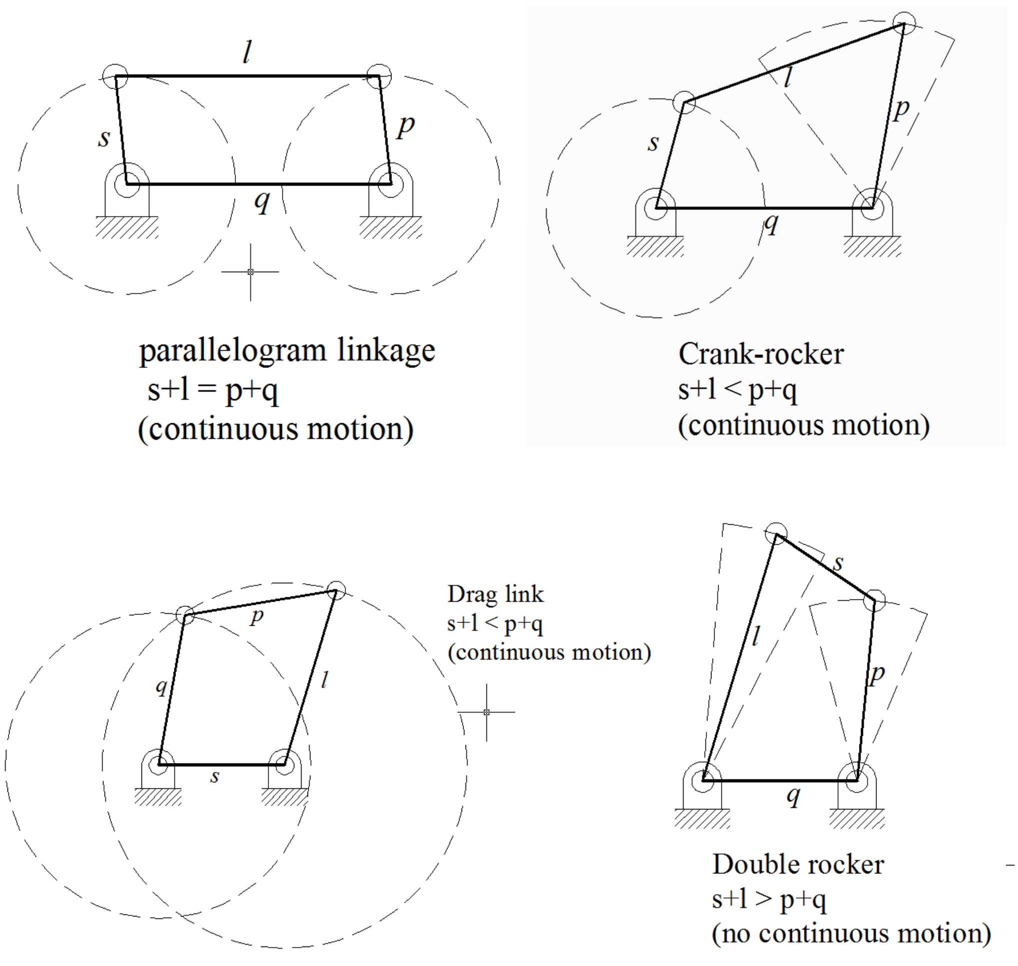


Figure 1.1. Possible Four bar configurations

(ii) *Case 2: $l+s > p+q$*

The lengths of shortest and the longest links are greater than the sum of the lengths of the other links. One double-rocker mechanism that is four different mechanisms which depends on the fixed link.

(iii) *Case 3: $l+s = p+q$*

This condition gives four possibilities. The change point is important. At this point (center lines) all the links are collinear. At this intermediate position the direction of rotation may be changed by the follower linkage. A parellogram linkage is the special case. There is an equal opposite links (l+s=p+q). There is a double crank mechanism which is four possible position mechanism in this configuration. This is called as *double crank* and has problem about the condition of change point. A deltoid linkage is a special case at this position, two equal links are connected to other equal links. A crank rocker mechanism is obtainable as a frame with the longest link.

1.3. Function, Path and Motion Generation

There is a basic question that must be answered in the synthesis of mechanisms such that how can a rigid body in plane motion be guided through a number of specific positions by means of constrained mechanisms. The answer to this question can solve a great number of synthesis problems encountered in the design of mechanisms. Depending on the required kinematic characteristics to be satisfied by the design linkage, dimensional synthesis problems can be broadly classified as *motion generation*, *path generation* and *function generation*. [3, 5, 6]

(i) Motion Generation: a rigid body has to be guided in a prescribed manner in motion generation. Motion generation is related with links controlling the links in the plane. The link is required to follow some prescribed set of sequential positions and orientations.

(ii) Path Generation: If a point on floating link of the mechanism has to be guided along a prescribed path, then such a problem is classified as a problem of path generation. Path Generation controls the points that points follow any prescribed path.

(iii) Function Generation: The function parameters (displacement, velocity, acceleration etc.) of the output and input links are to be coordinated to as to satisfy a prescribed functional relationship. The Function Generation is related with functional relationship between the displacement of the input and output links. [7]

A four link mechanism is synthesized to coordinate three positions of input and output links. The Freudenstein equation permits us to design a four-link mechanism for three precision positions of input and output links. A four link mechanism can be designed which is at five precision positions. The synthesis equations are nonlinear the other approaches are required to solve such synthesis equations. Two different types of output motion are obtained. The follower link and the coupler link derive the output motions. In synthesing a six-link mechanism, the coupler curves are utilized that this is obtained at the programmed motion of its output link. [4]

Use of the coupler curves is useful for machine design problems. The coupler curves are used for approximating complex planar paths such as *circles and elipses*. Three types of solution methods are applied as graphical, analytical and with optimization. Here the study is basically performed on 1 dof mechanisms with 4 links for generating the desired paths. Different graphical techniques will be examined for synthesizing a mechanism for different prescribed positions of its input and output links. (a) Four link mechanism, and (b) a slider-crank mechanism is used to synthesize by these techniques. These are three position synthesis, four-position synthesis and five-or-more-position synthesis. [4, 5]

1.4. Statement of Problem

The field of mechanism is usually doing the optimization of mechanism. If the dynamic characteristics of a mechanism is increased for safety, a multidiciplinary design optimization procedure is applied to synthesize optimum mechanisms. There are some input data for the algorithm that the designer needs these ones. These are DOF, range of all geometric parameters, inputs to the mechanism, outputs from the mechanism and the desired kinematic characteristics of the mechanism.

The objective is to apply an evolutionary method for synthesis of planar mechanisms and present a design guide for its use in linkage mechanisms. The evolutionary process is not related with the results which is obtained from enumeration of mechanisms. There is a key question for preparing an automated synthesis process which is representation. The mechanism can be encoded and all possible mechanisms can be desribed.

To solve this question, all possible mechanisms are described and all variation operators are used that to explore this space of mechanisms. A graph can be used as a representation of a kinematic mechanism that these are embedded in two or three dimensions. The links are at the edges of the graph and the joints are at the nodes of the graph. More this permits the genetic algorithm to generate task-spesific mechanisms. By using simulation the analysis engine evaluates the mechanism's performance then converts it to a standardized form which is usable by the synthesizer or any algorithm like GA. If the designer study on GA, one has to know the size of initial population, the selection scheme, the probability of crossover and mutation, the stopping criteria and other parameters.[7, 8]

In this study, there is a transformation in designing that show the kinematic functional requirements to mechanism models. In this study given as the mechanism topology and geometric parameters, we have to search for the optimum design. For the optimum design we get help from a GA. A new method for the automated kinematic synthesis of planar mechanism, the multibody system is given. The topology of pin connected planar mechanism is encoded in redesigning the two dimensional binary strings. There are some specific parameters which are related geometric parameters, initial conditions, task spesific parameters. These are redesigned to real number strings. The GA is used for searching for feasible topologies in different domain and optimum dimensions in a continuous domain. If we combine it with a multibody analysis package, GA can manage to generate mechanism designs greatly, maching the desired kinematic characteristics.[7]

Some algorithms are included in Matlab as toolbox facility. Genetic Algorithm and Neural Networks toolboxes are available. There is also Optimization toolbox where someone carry out optimization methods by looking at the problem chosen. Depending on the problem, preferably known routes will be utilized. Other available packages on mechanism synthesis will be looked for also.

1.5. Thesis Structure

This thesis includes five chapters on the research. It is organized as follows; First chapter outlines an introduction with synthesis of planar mechanism, statement of problem with the thesis structure. The necessary back ground is included on Evolutionary Algorithms used synthesis in Chapter 2. Literature survey is also given on mechanism synthesis using Evolutionary Algorithms (EAs) and it is ended with remarks on previous studies. Chapter 3 summarizes different optimization methods. Matlab Optimization Toolbox is introduced. Genetic algorithm Toolbox is also introduced. Some examples are given by using two optimization approach based on Matlab environment. Chapter 4 gives some illustrative examples on optimization based synthesis problems for 4 bar mechanism. Different mechanism examples are defined and their optimum solutions are discussed. Matlab Optimization Toolbox with constrained optimization is compared with Genetic Algorithm Toolbox (GA). Chapter 5 offers conclusions on the subject and recommendations for future work in mechanism optimization.

CHAPTER 2
LITERATURE SURVEY

2.1. Evolutionary Algorithms

Evolutionary Computation (EC) includes wide variety of application areas as planning, design, simulation and identification, control and classification. Planning will be routing, scheduling and packing in real world problem applications. Many real world problems are solved by application of evolutionary algorithms(EA). These algorithms are getting popular solving complicated nonlinear problems in engineering. Many evolutionary algorithms (EAs) are used like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Ant Colony Optimization (ACO), Simulated Annealing (SA), Genetic Programming (GP), Shuffled Frog Leaging (SFL), Memetic Algorithm (MA) etc. Their several combinations are also applied by looking at their advantages and disadvantages. Some of the algorithms are based on swarm intelligence like P*article Swarm Optimization* (PSO) or inspired by the mechanism of natural selection like *Genetic Algorithm* (GA) or *Neural Networks* (NN). [9]

Simulation and identification include the cases taking a design or model for a system. EC can also be applied in control as off-line and on-line. When it is used on-line, Evolutionary Algorithm performs active control. Off-line cases just include the controller design. Evolution increases intelligence over time. Some research has been seen on classifier systems as the learning classifier system primarily for Genetic Algorithms (GAs). Similarly techniques based on swarm intelligence are possibly applied for solving various optimization problems in engineering [9, 10].

2.2. Genetic Algorithm (GA)

Genetic Algorithm (GA) is an optimization technique based on the principles of genetics and natural selection. GA uses analogy to chromosome encoding and natural selection. This method is developed by J. Holland (1975). GA cannot be the best way to solve the engineering problem of interest. This is also true for other optimization algorithms whether they are based on mathematics or biological basis. It is also used for training artificial neural networks (ANNs). GA begins by defining the optimization variables, *the objective function* and *the fitness*. The algorithm ends with testing its convergence as an acceptable solution. Fitness is the term used to describe the output of the objective function. Most optimization problems require constraints. Thus GA starts with a group of chromosomes known as '*the population*'. [11]

A flowchart for simple GA can be found in books on the subject everywhere. It is shown in Figure 2.1. There are three genetic operators used to perform any operation. The operators are selection, crossover and mutation.

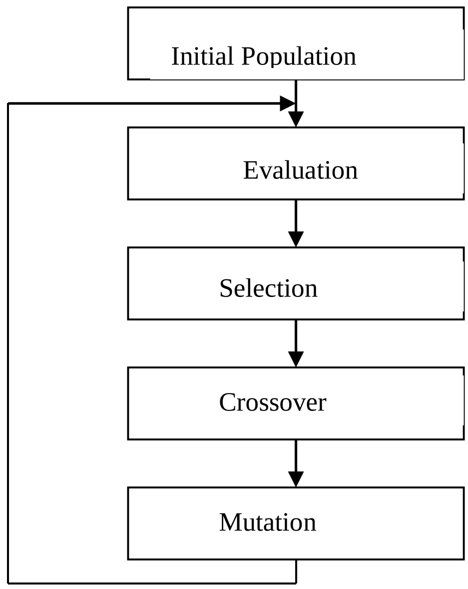


Figure 2.1. A Simple Genetic Algorithm

2.3. Survey on Synthesis of Planar Mechanisms

Many studies are seen on optimization based synthesis and optimization using evolutionary algorithms in literature. Some of them are directly related to the subject in concerned. They are included in the following part, and appeared with the years where the studies were performed.

J.S. Hoskins and G.A Kramer (1993) have previously introduced use of ANNs with optimization techniques (Levenberg-Marquardth Optimization) to synthesize a mechanical linkage generating a user-specified curve. A four bar planar linkage is studied to generate a coupler curve which is user specified. Inverse modeling is achieved using Radial Basis Function (RBF) ANN to find the linkage parameters also [12].

M.H.F.Dado and Y.S.Mannaa (1996) have described the principles for an automated planar mechanism dimensional synthesis. An algorithm is improved and presented for deriving the synthesis equations. The method is applied for a six bar and an eight bar mechanism [13].

R.C. Blackett (2001) has presented a technique for the optimal synthesis of planar five link mechanisms in Master's Study. A desired mechanical advantage function has produced on a specified path in the study. Hooke and Jees technique is applied to synthesize five-bar linkages [14].

P.S.Shiakolas and et al (2002) have presented representative examples utilizing Matlab through a web browser interface. Simulations can be carried in real time, and synthesize and analyze of planar mechanisms can be performed with knowledge of the analysis software. Motion generation is applied such that control of a line in a plane is achieved to perform prescribed precision positions [15].

J. A Cabrera et al. (2002) have dealt with solution methods of optimal synthesis of planar mechanisms. Genetic algorithms are applied and three examples are included in the study. The starting population is taken as sets of design variables [16].

R. Bulatoviç and S.R Djordjevic (2004) have performed optimal synthesis of four bar linkage by method of controlled deviation with Hooke-Jeeves's optimization algorithm. This is shown by using a four bar linkage. The coupler point is passed using 10 points on a straight line. Then the method is illustrated on the straight line path [17].

Laribi et al (2004) have proposed a combined Genetic algorithm- Fuzzy Logic Method (GA-FL) to solve the problem of path generation in mechanism synthesis. Several examples are presented in Matlab. A four bar mechanism is used for path generation [18].

K.G.Cheetancheri and H.H.Cheng (2005) have presented a study on computer aided analysis of mechanisms using Ch Excel. Script computing was performed with Ch, an embedded able C/C+ + interpreter. Mechanism design and analysis modules were developed in Ch, Ch Excel and Ch Mechanism Toolkit for four bar linkage. Other different mechanisms; the geared five bar, the crank slider and the multi-loop six bar linkages are also studied [19].

J.F. Collard et all have presented (2005) a simple approach to optimize the dimensions and the positions of 2d mechanisms for path or function generator synthesis. Two optimization strategies are developed and presented with three applications; a simple four bar path synthesis, a four bar steering linkage synthesis for function generation and a six bar steering mechanism [20].

H.H.Cheng and D.T.Trang (2006) have presented a study on a web-based mechanism analysis and animation. In the study, special four bar mechanisms as Grashof, non-Grashof, straight-line, quick return, and symmetrical linkages are included. The system is the web-based with graphical features to simulate the motion of given mechanism also. The web-based system is used to perform kinematics and kinetic analysis of different mechanisms as four bar, crank-slider, geared five bar, six bar linkages and cam follower systems [21].

J. Xie and Y.Chen (2007) have proposed an approach to kinematics synthesis of a crank rocker mechanism. Coupler link motions passing from a prescribed set of positions are generated. Artificial Neural Networks (ANN) is used to characterize the relationship between the mechanism dimensions and Fourier transformation. So a kinematic mapping is performed for desired set of positions of coupler link with Back Propagation Neural Network (BP-NN) [22].

Y.Liu et al (2007) have presented a new approach using the framework of genetic algorithms (GAs). Both mechanism topology and geometric parameters have been used for the optimum design with GA. Topology information contains which link is connected to the other with what kind of joint. Application of the algorithm for path generation problems is also given.[23]

S.Erkaya and İ Uzmay (2008) have presented a study on dimensional synthesis for a four bar path generation with clearance in joints. A method based on NN-GA is proposed and the joint characteristics with clearance are defined. The design variables are found at optimum to decrease the transmission angle in the mechanism [24].

N.N. Zadeh et al (2009) have used hybrid multi-objective genetic algorithms for Pareto optimum synthesis of fur-bar linkages. Objective functions are taken tracking error (TE) and transmission angles deviation (TA). They have compared optimum points with iterative studies. Two case studies were given. [25]

S.K. Archaryya and M. Mandal (2009) have performed a study on performance of EAs for four-bar linkage synthesis. Three different evolutionary algorithms such as Genetic algorithm (GA), Particle Swarm Optimization (PSO), differential Evolution (DE) have been applied for synthesis of a four bar mechanism. The error between desired and obtained coupler curves is minimized. Three algorithms are compared in terms of their performance at the end.[26]

A. Kentli et al. (2009) have presented a study on genetic coding application (GCA) to synthesis of planar mechanisms. The differences between GA and GCA are given. Some disadvantages seen in GA are eliminated by using GCA. Methods are compared using a case study based on a four bar mechanism [27].

K. Sedlaczek and P. Eberhard (2005) have presented a study on extended Particle Swarm Optimization technique based on the Augmented Lagrangian Multiplier Method. This method is called Augmented Lagrangian Particle Swarm Optimizer (ALPSO) and dimensional synthesis of a slider-crank mechanism is designed to follow an elliptic target trajectory as required. It is applied for synthesizing a slider crank mechanism with workspace constraints. The results are presented with 200 particles and 500 iterations. [28]

F. Penunuri et al (2011) have given optimal dimensional synthesis for planar mechanisms using differential evolution (DE) with four examples. The synthesis of a mechanism with hybrid tasks including path, function and motion generation was included. Classical DE algorithm was explained with mechanism synthesis problem. [29]

2.4. Survey on Synthesis of Hybrid Machines and Mechanisms

A hybrid machine is a machine which includes two actuators; a servo motor and a constant speed motor to drive a mechanism Tokuz (1992) was proposed the hybrid machine with a slider crank configuration to produce reciprocating motion. Modeling and control issues were studied, but synthesis work on the slider crank is not performed in the study. [30]

J.D. Greenough et al (1995), have studied on design of hybrid machines. A detailed study was presented for adjustable mechanisms with 2 dof. A five bar mechanism and a seven bar mechanism were applied. The dimensional design parameters are optimized in the kinematic synthesis.[31]

M.A.Connor et al (1995), have presented a methodology for the synthesis of multi degree of freedom mechanisms using Genetic Algorithms. GA was used to find the optimum link lengths minimizing the objective function. This was applied to a five bar mechanism where one of the input was provided by a constant speed motor and the other was supplied by a servo motor. [32]

H Zhou and E.H.M. Cheung (2000), have studied on analysis and optimal synthesis of hybrid five bar linkages for trajectory generation. A modified genetic algorithm was applied for the optimal solution. Motion range of hybrid five bar linkages was studied, and optimal synthesis was applied. Programmable output motion was generated with this configuration. Flexibility depends on the type and dimensions of hybrid mechanism chosen. [33]

A Kirecci and L.C.Dülger, (2000) have proposed a configuration with 2 dof, seven link mechanism with an adjustable crank. Chosen configuration was used for implementing several trajectories with power requirements as well. Many different linkage configurations were tried, and a seven bar linkage was replaced instead of a differential gear box. [34] A Kirecci and L.C.Dülger, (2003) have presented a study on motion drive and implementation for a hybrid drive system with seven link configuration. The problem of practical implementation was seen in an experimental set up without optimization for linkage parameters. Tracing errors were given at the end for the configuration. [35]

K Zhang, (2008) has studied on synthesis of a hybrid five mechanism with Particle Swarm Optimization (PSO) algorithm. The configuration is a parallel robot combining the motions of two characteristically different motors to get a programmable output. Optimal dimensional synthesis was described for hybrid five bar mechanism with related kinematics and dynamics. PSO based algorithms was also applied. [36]

2.5. Computational Methods in Mechanism Synthesis

Mechanism design and analysis have performed by using computational methods. Many complicated mechanisms can be designed. There are various general and special purpose packages available for this purpose. General purpose packages as Automated Dynamic Analysis of Mechanical Systems (ADAMS), Dynamic Analysis and Design System (DADS), Working Model, Pro/Engineer were mentioned. Special purpose packages as the Linkage Interactive Computer Analysis and Graphical Enhanced Synthesis Package (LINCAGES), WATT, Simulation and Analysis of Mechanisms (SAM) are found for the synthesis and analysis of planar mechanisms. There is another software package SYNTHETICA for synthesis of spatial mechanisms also. [37]

2.6. Remarks on Literature Survey

Different techniques have been applied for synthesis of planar mechanisms. The development of computer routines was made possible and easy to used numerical methods to the minimization of objective function. There is a special need of high effort for optimization in synthesis of planar mechanisms. Many attempts are seen to apply artificial intelligence techniques. Most of the analyses are carried on planar mechanisms.

Many optimization algorithms are applied. Among them PSO, GA, DE are the ones commonly seen. Evolutionary algorithms have been applied to get the optimum. Most of the studies, four bar mechanism is chosen. Since it is a simple mechanism and applicable to many different applications. This study also presents a study on four bar mechanism for its further use. Constrained optimization study is carried out using Matlab with Genetic Algorithm during study. Optimization toolbox is explored with GA Toolbox.

CHAPTER 3
OPTIMIZATION ALGORITHMS

3.1 Introduction

Optimization methods are used in industry, computer science, business and engineering. During the last 10–15 years mathematical programming advanced from an interdisciplinary problem-oriented approach to an independent, rigorously developed branch of applied mathematics. This development, was certainly desirable. It has caused a certain polarization between theoreticians working in a research environment and practitioners in industry.

Engineers in the past were actively involved both in the modelling of optimization problems and in the development of suitable algorithms for their solution. Engineers who have to be familiar with complicated and rapidly changing technologies, can not simultaneously be experts in modern techniques of mathematical programming. It is therefore, conceivable that as in other applications of optimization techniques like planning, engineering optimization must be carried out with the aid of mathematical programming Here the engineer takes role of providing the necessary back ground material for model formulation. They certainly participate in the solution process by examining and verifying the results.

The main purpose of this study is to bring to the attention of mathematical programmers a selection of engineering optimization problems chosen from several disciplines and environments. Optimization is the process of making something better. An engineer or scientists takes up a new idea and optimization improves on that idea. Optimization consists in trying variations on an initial concept and using the information gained to improve on the idea. This chapter gives information on engineering optimization andalso Matlab Optimization Toolbox with some explicit examples.

3.2 Engineering Optimization

Engineering optimization introduces as a way of finding a set of parameters that can in some way that is defined as optimal. These parameters are found from objective function which are minimized or maximized and subject to equality or inequality constraints and/or parameter bounds. This terminology "best" solution implies that there is more than one solution and the solutions are not equal value. The optimum solution depends on the person formulating the problem. Some problems have exact answers or roots, and the best has a spesific definition. Examples include solution to a linear first-order differential equation. Other problems have various minimum or maximum solutions known as optimal points or extrema, and the best may be a relative definition. Science developed simple models to represent certain limited aspect of nature. Most of these simple (and usually linear) models have been optimized. In the future, scientists and engineers must tackle the unsolvable problems of the past, and optimization is a primary tool needed in the intellectual toolbox [38].

Bracketing a minimum requires three points, with the middle point having a lower value than either end point. In the mathematical approach, root finding searches for zero of a function, while optimization finds zeros of the function derivative. Finding the function derivative adds one more step to optimization process. Many times the derivative does not exist or is very difficult to find. Another difficulty with optimization is determinig if a given minimum is the best (global) minimum or suboptimal (local) minimum. Finding the minimum of nonlinear function is espectially difficult. Typical approaches to highly nonlinear problems involve either linearizing the problem in a very confined region or restricting the optimization to a small region. In the processof optimization, a real problem is defined with its details. A model is built and an algorithm or a solution method is chosen and applied. Then the computer carries out the calculation necessary. This process is given in Figure 3.1.

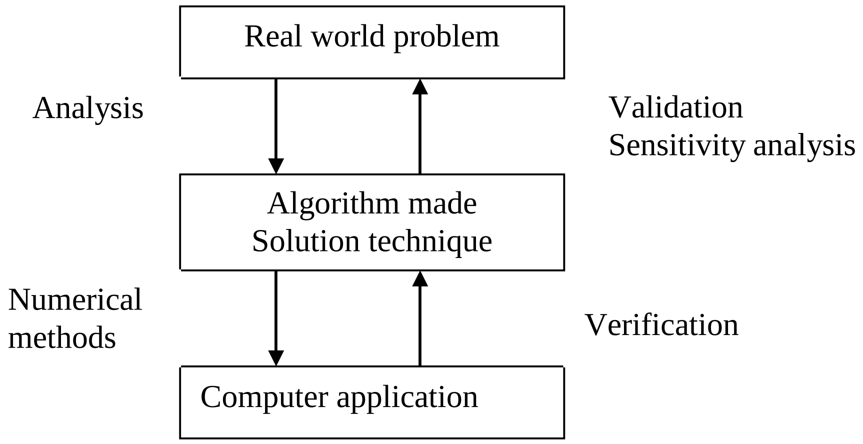


Figure 3.1. Process of optimization

3.3. Matlab Optimization Toolbox

Optimization is the process of adjusting the inputs to or characteristics of a device, mathematical process, experiment to find the minimum or maximum outputs or result. The input consists of variables; the process of function is known as *the cost function*, the *objective function*, or *the fitness function*; and the output is the cost or fitness. If the process is an experiment, then the variables are physical inputs to the experiment.

Many different functions are collected in matlab computing environment. Many types of optimization routines are included. As we know that this is to find the x value of function which is the found from first derivative and equal to zero value. The second derivative defines whether it is minimum or maximum points. Optimization concerns the minimization or maximization of functions. For solving the optimization problems, f(x) curve is searched and a decision is made on the way to go.

There are three different programming such as;

(i) f(x) and the constraints are linear, the linear programming is applied.

(ii) f(x) is second order, the constraints are linear, the quadratic programming is applied.

(iii) f(x) is not linear, second order, and the constraints are not linear, nonlinear programming is applied.

While the variations in the problem that obtains the optimum point and it must be stated at the interval points, this is named as *'constrained optimization'*. If there is not any constraints, this is called as *'unconstrained optimization*.'In Matlab, the methods are giveen using three diffeent topics as L- Large scale method, M- Medium scale method and B- Large and Medium scale method.

It is the collection of functions that extend the capability of MATLAB numerical computing environment. Optimization toolbox includes many types of optimization algorithms. The toolbox includes function for linear programming, quadratic programming, binary integer programming, nonlinear optimization, nonlinear least squares, systems of nonlinear equations, multiobjective optimization. It is used to find optimum solutions. These are given as:

- Unconstrained nonlinear minimization.
- Constrained nonlinear minimization, including goal attainment problems, minimax problems and semi-infinite minimization problems.
- Quadratic and linear programming.
- Nonlinear least squares and curve-fitting.
- Nonlinear system of equation solving.
- Constrained linear least squares.
- Sparse and structured large-scale problems.

All of the toolbox functions are MATLAB M-files, made up of Matlab statements that implement specialized optimization algorithms. The MATLAB code is written for these functions using the statement as **type function_name.** Matlab Optimization Toolbox includes Standard Algorithms and Large scale algorithms. These algorithms solve constrained and unconstrained continuous and discrete problems. Special techniques are needed to have a solution for large-scale optimization problems. Trust Region Methods for Nonlinear Minimization (TRMNM), Preconditioned Conjugate

Gradients (PCG), Linearly Constraint Problems (LCP), Nonlinear Least Squares (NLS), Quadratic Programming (QP), Linear Least Large-Scale Linear Programming (LLLSLP) [39, 40].

3.3.1. Standard Algorithms

The Standard Algorithm is organized into some subparts are unconstrained optimization, Quasi-Newton implementation, Least-squares optimization, Nonlinear least squares implementation, Constrained optimization, SQP implementation, Multiobjective optimization.

***(i) Unconstrained Optimization*:**

It is known as quasi-Newton and line search method. This provides implementation details for the Hessian update and line search phases of the quasi-Newton algorithm used in *fminunc.*

(ii) Least Squares Optimization:

For nonlinear least squares optimization there is a discussion for the use of the Gauss-Newton and Levenberg-Marquardt methods. Here the optimization routines *lsqnonlin & lsqcurvefit,* the Gauss-Newton and Levenberg-Marquardt methods are used.

***(iii) Constrained Optimization*:**

Sequential Quadratic Programming (SQP) methods, the use of the Kuhn-Tucker (KT) equations is explained. SQP Implementation details for Hessian matrix update, quadratic programming problem solution, and line search and merit function calculation phases of SQP algorithm which is used in *fmincon, fminimax, fgoalattain* and *fseminf.*

(iv) Multiobjective Optimization:

This is an introduction for multiobjective optimization and discussion strategies for dealing with competing objectives. It suggests improvement to the SQP method and discussed the use of goal attainment method for use with the goal attainment method. These support the concept implemented in the medium-scale algorithms.

3.4. Optimization with Genetic Algorithm (GA)

The Genetic Algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. GA allows a population composed of many individuals to envolve under specified selection rules to a state that maximizes the fitness (i.e., minimizes the cost function). The method was developed by John Holland(1975) over the course of the 1960s and 1970s and finally popularized by one of his students. Since then, many versions of evolutionary programming have been tried with varying degrees of success. There are many advantages in its use [11]; they can be summarized as

- Optimized with continuous or discrete variables,
- Simultaneously searches from a wide sampling of the cost surface,
- Deals with a large number of variables,
- Optimized variables with extremely complex cost surfaces (they can jump out of local minimum.)
- Provides a list of optimum variables, not just a single solution,
- the optimization is done with the encoded variables,
- Works with the numerically generated data, the experimental data, or the analytical functions.

GA is not the best way to solve every problem. For instance, the traditional methods have been tuned to quickly find the solution of a well-behaved convex analytical function of only a few variables. For such cases the calculus-based methods out perform the GA, quickly finding the minimum while the GA is still analyzing the cost of the initial population. For these problems the optimizer should use the experience of the past end employ these quick methods. However many realistic problems do not fall in to this category. In addition, for problems that are not overly difficult, other methods may find the solution than the GA. The large population of the solutions that gives the GA its power is also its bane when it comes to speed on a serial computer- the cost function of each of those solutions must be evaluated. GA is optimally suited for such parallel computation [40, 41].

3.4.1. Genetic Algorithm Toolbox

The Genetic Algorithm GUI Toolbox is very important. By typing simply optimtool (‘ga’) in the command prompt, then the GA toolbox becomes active. The main data structures in GA Toolbox are chromosomes, the objective function values and the fitness values. The implementation of the Genetic algorithm with Matlab is given. The functions; objective, crossover operations, mutation operations, fitness scaling are given. They are listed for implementing optimization [42, 43].

3.5. Optimization Examples

This section includes some optimization examples using Optimization Toolbox with comments. Two examples are given to show nonlinear constrained optimization. *fmincon* represents multidimensional constrained nonlinear minimization. During the studu beforestarting to write our own functions, some example functions are studied and presented as tutorial. Mechanism based examples; especially for four bar mechanism are included in Chapter 4.

3.5.1 Using *fmincon* in MATLAB

<u>Example 1:</u> First example shows how to use the optimization tool with the *fmincon* solver to minimize a quadratic subject to linear and nonlinear constraints and bounds. Consider the problem of finding $[x_1, x_2]$that solves $\min f(x) = x_1^2 + x_2^2$ *such that x* subject to the constraints $0.5 \le x_1$(bound). The starting guess for this problem is taken as x_1=3 and x_2=1.

$-x_1 - x_2 + 1 \le 0$ (linear inequality)

$-x_1^2 - x_2^2 + 1 \le 0$ (nonlinear inequality)

$$-9x_1^2 - x_2^2 + 9 \le 0 \qquad \text{(nonlinear inequality)}$$

$$-x_1^2 + x_2 \le 0 \qquad \text{(nonlinear inequality)}$$

$$-x_2^2 + x_1 \le 0 \qquad \text{(nonlinear inequality)}$$

Table 3.1. M-file showing the first example

Step 1: Write an M-file objfun.m for the objective function.

```
function f=objfun(x)
f=x(1)^2+x(2)^2;
```

Step 2: Write an M-file nonlconstr.m for the nonlinear constraints.

```
function [c,ceq]=nonlconstr(x)
c=[-x(1)^2-x(2)^2+1;
    -9*x(1)-x(2)^2+9;
    -x(1)^2+x(2);
    -x(2)^2+x(1)]; ceq=[];
```

Step 3 : Set up and run the problem with the Optimization Tool.

Enter *optimtool* in the command window to activate the optimization tool.

Select *fmincon* from the selection of solvers, change the **Algorithm** field to *Active set*.

The optimization problem is defined, and The Optimization tool is started. O*bjfun* is entered in the **objective function** field to call the M-file *objfun.m*. Enter [3;1] in the start point field as seen in Figure 3.2.

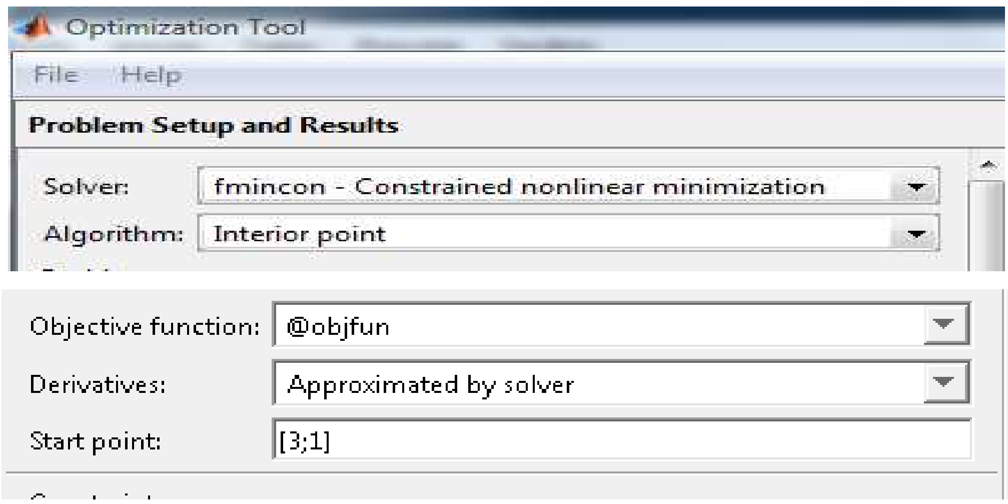


Figure 3.2. Solver *fmincon* window (*the first example*)

The constraints are set. The bound is given as $0{,}5 \leq x_1$ by entering 0,5 in the **lower** field. The linear inequality constraint is then set by entering [-1 -1] in the **A** field and enter -1 in the **b** field. Next the nonlinear constraints are typed by entering nonlconstr in the **Nonlinear constraint function** field as shown in Figure 3.3.

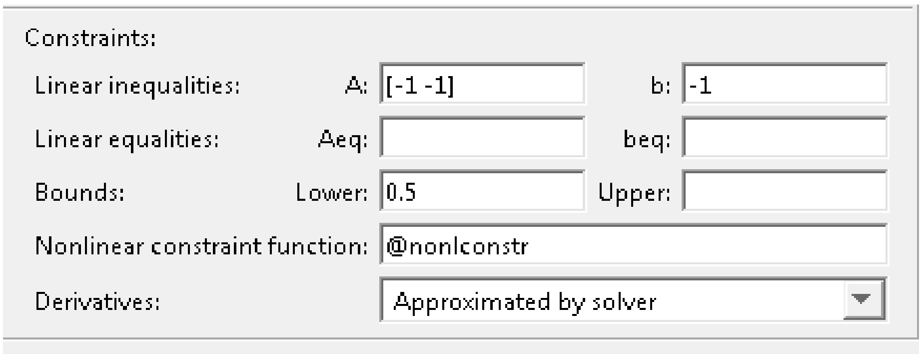


Figure 3.3. Constraints applied (1st example)

The start button is clicked to run the program, and the **run solver and view results** window is activated, the following information is displayed like in Figure 3.4.

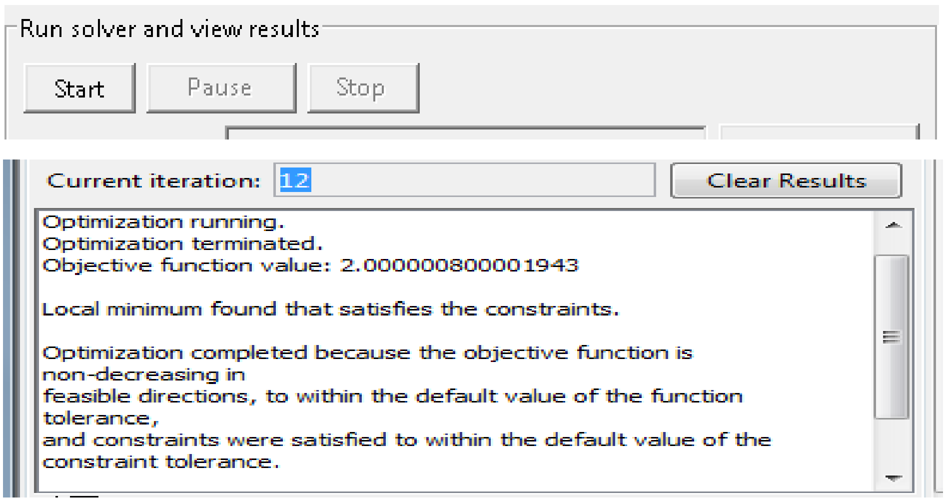


Figure 3.4. Status and Optimization result (*fmincon-1st example*)

The current iteration value when the algorithm terminated, which for this example is 12. The final value of the objective function when the algorithm terminated. Objective function value is found as 2.000000800001943.

Example 2: This example shows how to use minimization between the two lines that they are defined as a_1= 6 unit and a_2= 1 unit. Table 3.2.shows necessary M-file for this optimization problem.

$x_1 = \cos\theta$ $\qquad\qquad$ $x_2 = \sin\theta$

The problem is considered finding x1, x_2 and x_s, y_s solves;

$$\min f_x(x) = \sqrt{(x_s^2 + y_s^2)}$$

$x_s = -a_1 + a_2 x_1$ $\qquad$ and $\qquad$ $y_s = a_2 x_2$

It is subjected to the constraints where $-1 \le x_1 \le +1$ and $-1 \le x_2 \le +1$ (Bound)

$-7 \le -a_1 + a_2 x_1 \le -6$ $\qquad$ (linear inequality)

$-1 \le a_2 x_2 \le +1$ $\qquad$ (linear inequality)

Table 3.2. M-file showing the second example

Step 1: Write an M-file objfun2.m for the objective function.

```
function f=objfun2(x)
f= [36-12*x(1)+x(2)^2+x(2)^2]^0.5
```

Step 2: Write an M-file nonlconstrl2.m for the nonlinear constraints.

```
function [c,ceq]=nonlconstrl2(x)
c=-6+x(1);
     x(2)
     ceq=[];
```

Step 3 : Set up and run the problem with the Optimization Tool.

Enter *optimtool* in the command Window to open the optimization tool.

Select *fmincon* from the selection of solvers and change the **Algorithm** field to *Active set.*

Enter *objfun* in the **objective function** field to call the M-file *objfun2.m*. Then enter [0;1] in the starting point field like in Figure 3.5.

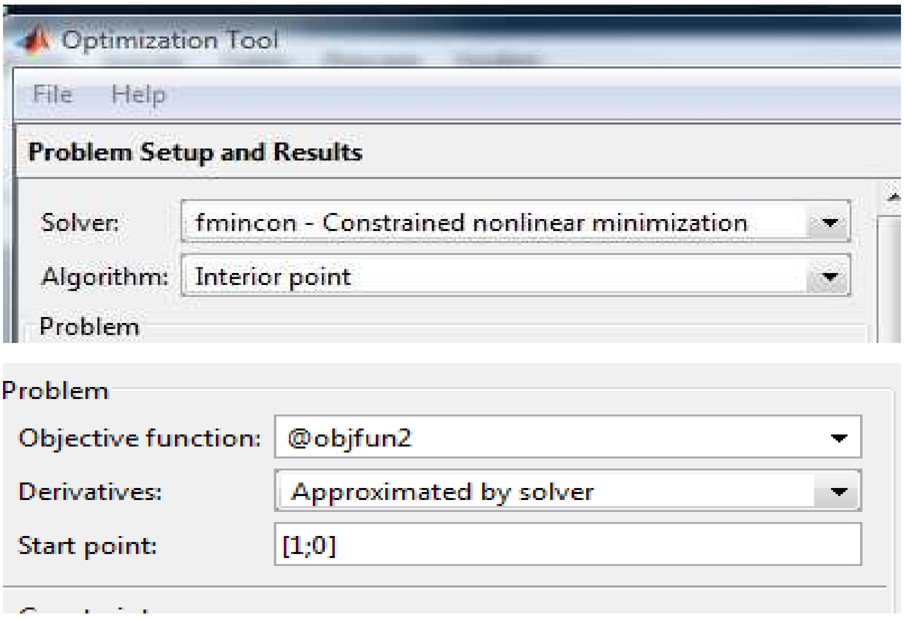


Figure 3.5. Optimization Tool Status (*the second example*)

The constraints are defined. The nonlinear constraints are by entering *nonlconstrl2* in the **Nonlinear constraint function** field. The lower and upper bounds are set to -7 and +1 respectively. Then the start button is clicked to run the program. When the **run solver and view results** is activated, the following information is displayed like in Figure 3.6.

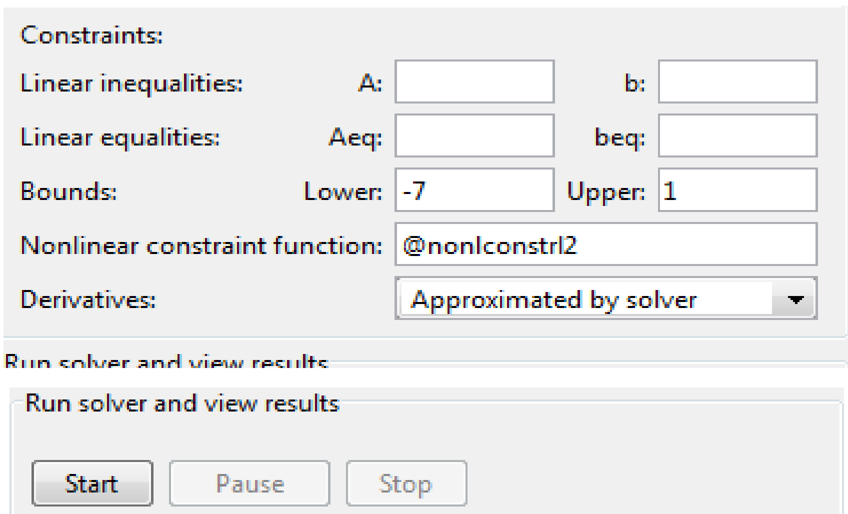


Figure.3.6. Constraints applied (*the second example*)

The **current iteration** value when the algorithm terminated, which for this example is 10. The final value of the objective function when the algorithm terminated.

Objective function value is found 4.898979505566125 as shown ,n
Figure 3.7.

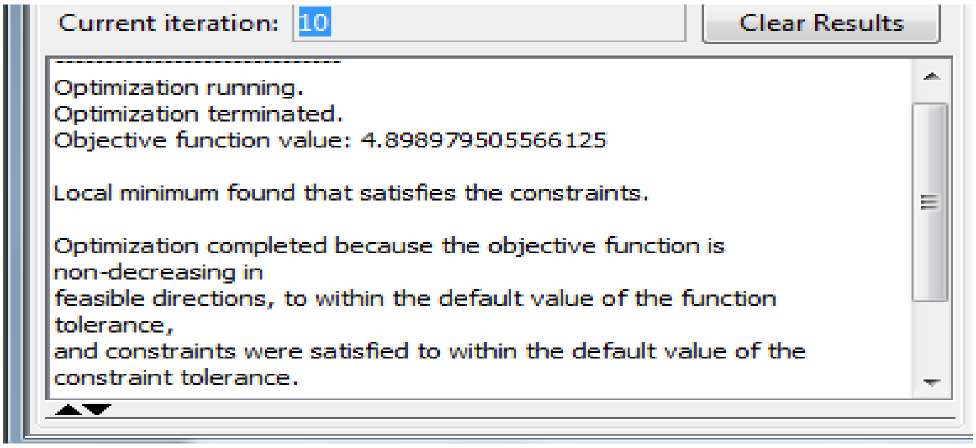


Figure 3.7. Status and Optimization results(*fmincon-the second example*)

3.5.2 Using *Genetic Algorithm (GA)* in MATLAB

Example 1 which is given previously is resolved by using Genetic Algorithm methods. The objective function is f = objfun (x) where $f = x_1^2 + x_2^2$ is shown in Figure 3.8. The M-file nonlconstr.m is used for nonlinear constraints is given in Figure 3.9.

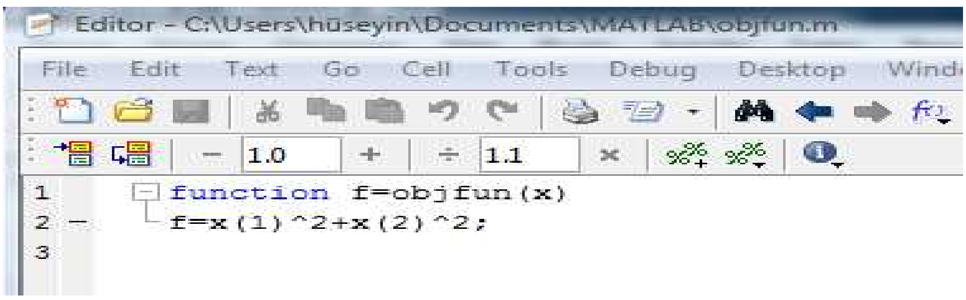


Figure 3.8. Function m.file in Matlab (1st example)

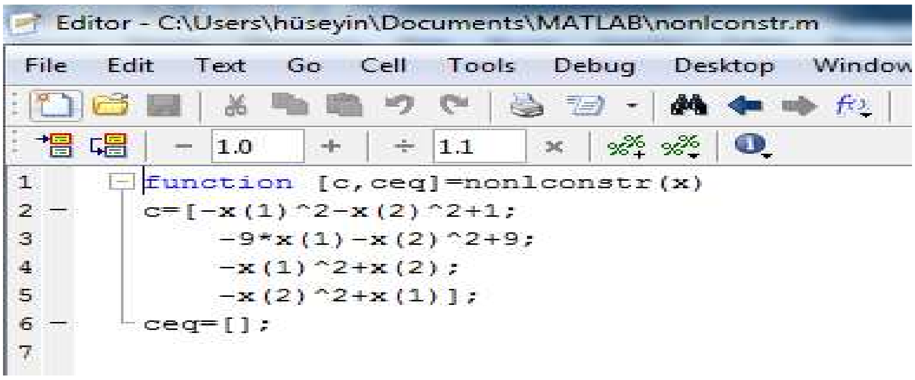


Figure 3.9. Constraint function m-file in Matlab (1st example)

By using the command line interface, the GA can run many times with different option settings as shown in Figure 3.10..

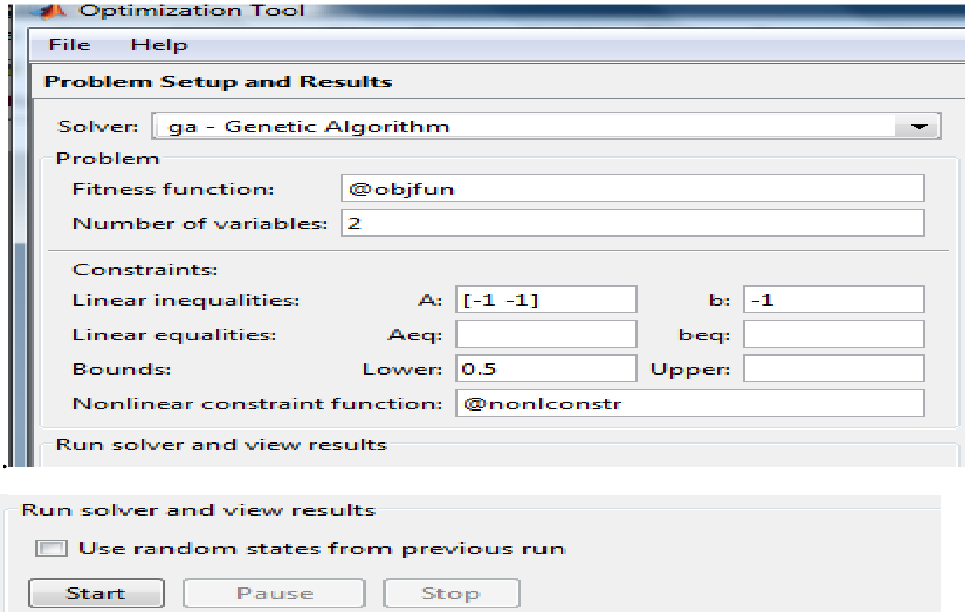


Figure 3.10.Optimization tool status (*GA-first example*)

The optimtool is provoked as shown here; then the start buton is activated. The progress of the optimization can be seen on the screen in Figure 3.11.

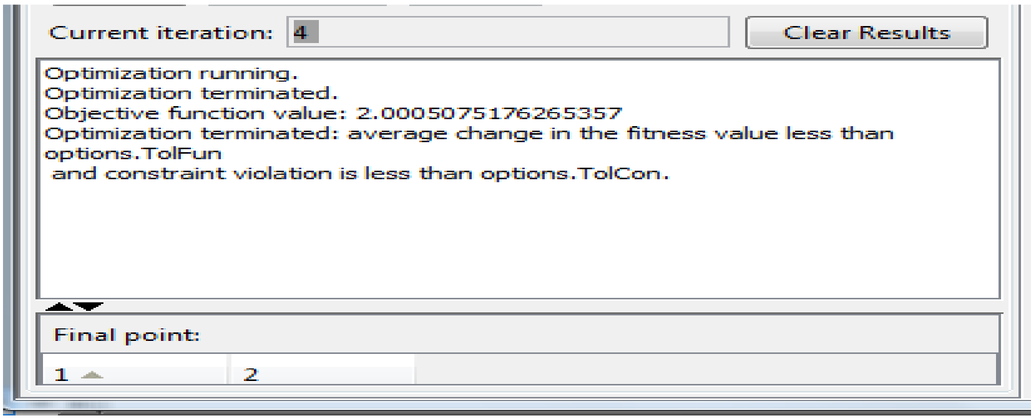


Figure 3.11. Optimization result (*GA-the first example)*

Using Genetic Algorithm (GA); the objective function value is found as 2.0005075176265357**.** This result can be checked using the optimization toolbox (fmincon) the objective function value is seen as 2.000000800001943**.** These two values are almost equal to each other.

Similarly Example 2 is resolved by using Genetic Algorithm. The objective function is function taken as f = objfun2 (x) and $\min f_x(x) = \sqrt{(x_s^2 + y_s^2)}$ The optimtool is typed as problem setup. Solver is attained as 'ga'. If GA converges to a solution, it is assumed that the nonlinear constraints are also satisfied. The screen is shown in Figure 3.12.

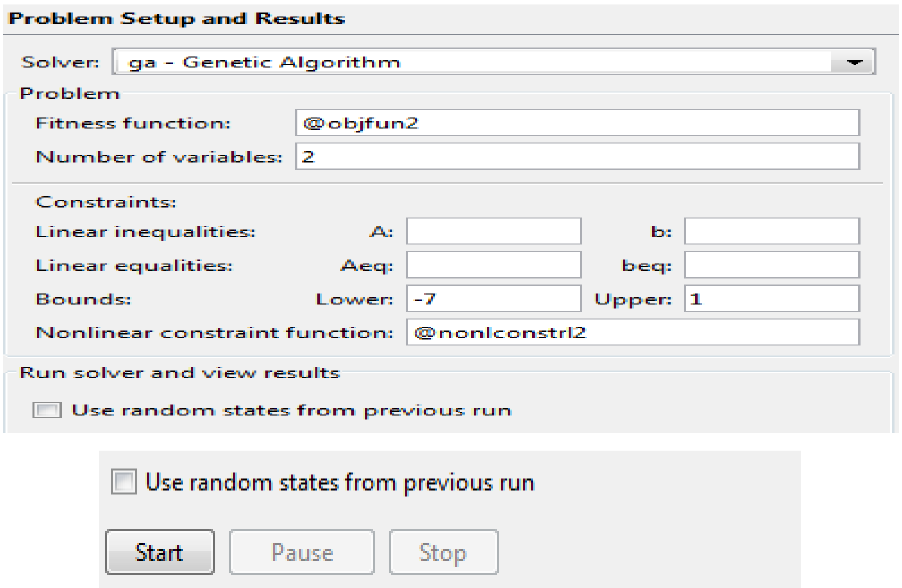


Figure 3.12. Optimization tool status (*the second example-GA*)

The start buton is activated and the following results are seen on the screen. It is shown in Figure 3.13.

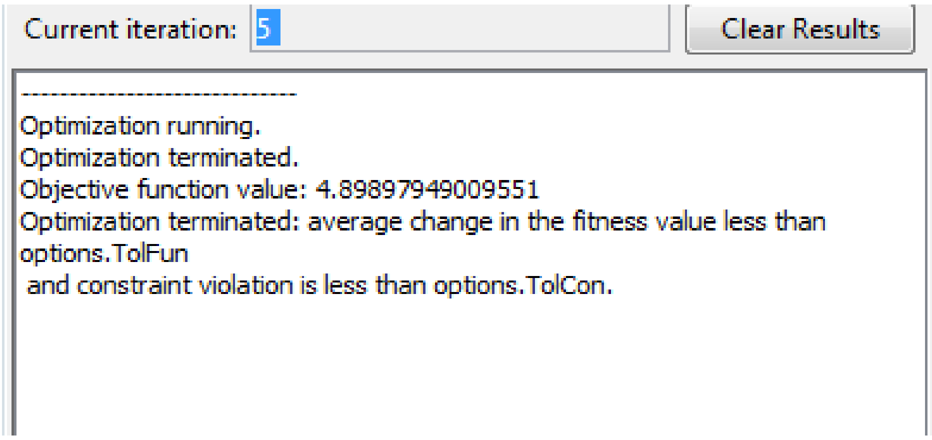


Figure 3.13. Status and Results (*the second example-GA*)

Genetic Algorithm (GA) is applied and the objective function value is 4.89897949009551**.** The optimization toolbox is given the objective function as 4.898979505566125. These two values are almost equal to each other.

CHAPTER 4
OPTIMUM SYNTHESIS OF PLANAR MECHANISMS

4.1. Introduction

Analysis of mechanisms is the study of motion of different members constituting a mechanism and the mechanism as a whole entity while it is being operated or run. This study of motion involves linear as well as angular position. velocity and acceleration of different points on members of mechanisms. There are two methods and techniques for the analysis of mechanisms which are graphical method and analytical method. The motion characteristics of the known mechanisms will be determined. General methods for determination of motion. velocity and acceleration of mechanisms will be given.

'Motion study" is a catch-all term of simulating and analyzing the movement of mechanical assemblies and mechanisms. The continuous contribution by design engineers for years has lead to development of many methods and techniques for analysis of mechanisms. Recently. the development of computer techniques have offered a number of viable and attractive solutions. Thischapter utilizes a simple but important mechanism; four bar for sythesis application.

Initially kinematic analysis are performed. This mechanism is optimized using multidimensional constraint optimization. Matlab based Optimization Toolbox is utilized initially. Then GA toolbox is applied for GA based optimization offering one of the heuristic approaches. Different case studies are explored and compared to show how effective both algorithms are.

4.2. Kinematics of Four Bar Mechanism

The kinematic analysis of a four-bar mechanism is considered first. Figure 4.1 shows four bar mechanism in general coordinate system [16. 26]. The design procedure of a four-bar linkage starts with the vector loop equation. which is referred to Figure 4.1. The position vectors are given as $\vec{R}_1, \vec{R}_2, \vec{R}_3, \vec{R}_4$; The offset angle is notated by θ_0 and the input angle is θ_2. The position vectors are used to get complete four bar linkage.

$$\vec{R}_2 + \vec{R}_3 = \vec{R}_1 + \vec{R}_4 \tag{4.1}$$

The complex number notation can be substituted next by using scalar lengths of the links as r_1. r_2. r_3 and r_4.

$$r_2 e^{i\theta_2} + r_3 e^{i\theta_3} = r_1 e^{i\theta_0} + r_4 e^{i\theta_4} \tag{4.2}$$

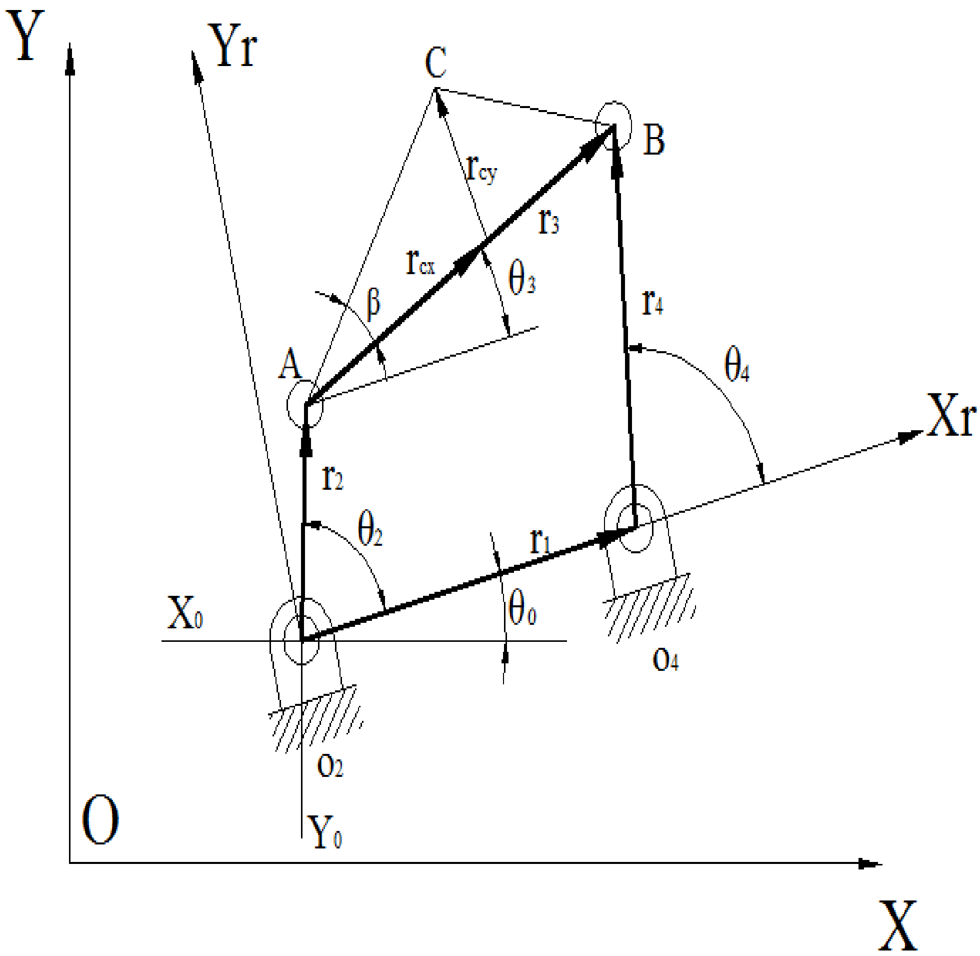


Figure 4.1. Four bar mechanism in general coordinate system.

Here θ_3 and θ_4 the angles to be found. They can be expressed as

$$\theta_3 = f\{r_1, r_2, r_3, r_4, \theta_2, \theta_0\}$$
$$\theta_4 = f\{r_1, r_2, r_3, r_4, \theta_2, \theta_0\}$$
(4.3)

Equation (4.2) is expressed with its real and imaginary parts with assumption of θ_0=0. then the real and imaginary parts are written as

$$r_2 \sin\theta_2 + r_3 \sin\theta_3 = r_4 \sin\theta_4 \qquad (4.4a)$$

$$r_2 \cos\theta_2 + r_3 \cos\theta_3 = r_1 + r_4 \cos\theta_4 \qquad (4.4b)$$

The unknowns are calculated by using Freudenstein's equation [3. 2.18]

$$K_1 \cos\theta_3 - K_4 \cos\theta_2 + K_5 = \cos(\theta_2 - \theta_3) \qquad (4.5a)$$

$$K_1 \cos\theta_4 - K_2 \cos\theta_2 + K_3 = \cos(\theta_2 - \theta_4) \qquad (4.5b)$$

K_1. K_2. K_3. K_4 and K_5 are found as

$$K_1 = \frac{r_1}{r_2}. \qquad K_2 = \frac{r_1}{r_4} \quad . \quad K_3 = \frac{r_2^2 - r_3^2 + r_4^2 + r_1^2}{2r_2r_4}. K_4 = \frac{r_1}{r_3}. \ K_5 = \frac{r_4^2 - r_1^2 - r_2^2 - r_3^2}{2r_2r_3}$$

The angles can then be found; (4.6)

$$\theta_{3_{(1,2)}} = 2\tan^{-1}\left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\right) \qquad (4.7)$$

$$\theta_{4_{(1,2)}} = 2\tan^{-1}\left(\frac{-E \pm \sqrt{E^2 - 4DF}}{2D}\right) \qquad (4.8)$$

In above equations; ± sign refers to two different configurations of the four bar mechanism. A. B. C. D. E and F expressions are then written as

$$
\begin{aligned}
A &= \cos\theta_2 - K_1 - K_2\cos\theta_2 + K_3 \\
B &= -2\sin\theta_2 \\
C &= K_1 - (K_2 + 1)\cos\theta_2 + K_5 \\
D &= \cos\theta_2 - K_1 + K_4\cos\theta_2 + K_5 \\
E &= -2\sin\theta_2 \\
F &= K_1 + (K_4 - 1)\cos\theta_2 + K_5
\end{aligned}
\tag{4.9}
$$

Again referring to Figure 4.1. the referance frame is taken as $O_2 X_r Y_r$. and the design variables for the mechanism are taken as $r_1, r_2, r_3, r_4, r_5, r_{cx}, r_{cy}, \theta_0, x_0 and y_0$. By taking $x_0 = 0, y_0 = 0, \theta_0 = 0$. the coupler position (C) can be written as

$$C_{xr} = r_2\cos\theta_2 + r_{cx}\cos\theta_3 - r_{cy}\sin\theta_3 \tag{4.10a}$$

$$C_{yr} = r_2\sin\theta_2 + r_{cx}\sin\theta_3 - r_{cy}\cos\theta_3 \tag{4.10b}$$

In previous notation. by taking OXY then;

$$\begin{bmatrix} C_x \\ C_y \end{bmatrix} = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{bmatrix} \begin{bmatrix} C_{xr} \\ C_{yr} \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{4.11}$$

Equation (4.11) is later used while performing derivations of the goal function for the mechanism.

4.3 Optimum Synthesis of Four Bar Mechanism

There is an increase in computer technology which has permitted us in developing routines that apply methods to the minimization of a goal function. There is a common goal function that is the error between the points tracked by the coupler (crank-rocker) and its desired trajectory in general. The aim is to minimize the goal function applying optimization techniques here.

4.3.1 Objective Function

Initially the link lengths are chosen according to the *Grashof's Theorem*. Many cases a continuous rotary input is applied and the mechanism must satisfy the Grashof criteria. The first part computes the position error in the objective function. The sum of the squares of the Euclidean distances between each point is defined and a set of target points indicated by the designer that should be met by the coupler of the mechanism. These points can be written in a global coordinate system as C_T^i are the target positions on the coupler.

$$C_T^i = \left[C_{xT}^i ; C_{yT}^i\right] \text{ where i=1.2.3.....n} \tag{4.12}$$

The variables can be optimized in case of problem without prescribed timing. Structural error is the error between the mathematical function and the actual mechanism. Cabrera et al [16] have defined a goal function. accordingly. the first part of goal function is expressed by minimize:

$$f_{obj} = \sum_{i=1}^{N} \left[\left(C_{xT}^i - C_x^i\right)^2 + \left(C_{yT}^i - C_y^i\right)^2\right] \tag{4.13}$$

where N is defined as the number of points to be synthesized.

The geometric magnitudes of four-bar mechanism are previously described in Figure 4.1. The design variables are $r_1, r_2, r_3, r_4, r_{cx}, r_{cy}$, and the input angle is θ_2. The second part of goal function is derived from the constraints which are imposed on the mechanism and set as:

- The Grashof condition allows for full rotation of at least one link.
- The sequence of input angles. θ_2 can be from the highest to the lowest (or from the lowest to the highest).
- The range for the design variables should be given.
- The range of variation for the input angle should be given.

The first three conditions are imposed and the fourth condition is taken as to perform full 360˚ rotation of the crank in the results presented here. In order to use this definition of the problem when the optimization algorithm is implemented. the set constraints are retained and the values are assigned to the design variables X.

4.4. Case Studies on Multiobjective Constrained Optimization

The objective function is constrained one for synthesizing four-bar mechanism. The Grashof 's condition and constraints regarding to sequential (CW or CCW) rotation of the input crank angle as written. The constraints play an important role in designing a feasible solution of the mechanism. A high number of initial populations are chosen randomly from the given set of minimum and maximum values of the variables so that a considerable amount of them can play in next iteration. This technique unnecessarily increases CPU time and reverses a large amount of memory in the computer. The refinement of population applied here is only for choosing an initial population and the other part of the evolutionary algorithms is kept same. The randomly chosen initial population is modified according to feasibility of making an effective mechanism.

In a randomly chosen variable set. the lengths of the linkage and the crank angle. θ_2 are taken. The linkage lengths are initially chosen as random. that may only satisfy the Grashof's condition. The lengths are then reassigned if they fail to satisfy this condition. After that randomly chosen. the input angles are rearranged in CW or CCW with randomly choosing first input angle among the initial generated set to meet the constraints. After these modifications in initial population, a comparatively greater number of strings can be found to make a feasible mechanism or the probability of rejection of strings in next iteration is reduced.

Here *fmincon* command is used for nonlinear and many variables. This is a gradient based search function in Matlab to solve the constraint problem. It is necessary to have a constrained m-file to run this program (main program) and to perform optimization. It is included in Appendix. Firstly the link lengths are defined as r_1, r_2, r_3, r_4 . The constraints are then defined according to the link lengths which are

related with the Grashof's Theorem to obtain a doublecrank mechanism (l-the longest link. s-the shortest link. p. q -two intermediate links as l+s ≤ p+q). So the link lengths are chosen according to these values as constraints. The parameters are set as
$l = r_1$ (the link 1). $s = r_2$ (the link 2). $p = r_3$ (the link 3). $q = r_4$ (the link 4)

4.4.1. Case I- Path generation without prescribed timing

There are six coupler points required to find out an optimal solution at the end. These points are designed to trace a vertical straight line by changing Y coordinate only. The problem is then defined by;

(i) The design variables are;

$X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_2]$ Where *i=1.2.….N and N=6*

(ii) Target points are chosen as:

$[C_{xT}^i, C_{yT}^i] = [(20,20),(20,25),(20,30),(20,35),(20,40),(20,45)]$

(iii) Limits of the variables;

$r_1, r_2, r_3, r_4 \in [13,70]$. $r_{cx}, r_{cy} \in [-60,60]$ and $\theta_2 \in [0,2\pi]$

where *i=1.2.….N and N=6*

(iv) Parameters of GA;

Population Size (PS) = 20. Crossover Possibility (CP) = 0.8. Mutation Possibility-uniform (MP)=0.1. Selection type=Roulette

(v) fmincon conditions;

Maximum iterations= 400

Optimization Toolbox command *fmincon* is compared with GA here. Both programs are run by following steps given in Chapter 3. M-files are written for four bar mechanism (see Appendix) and is called in optimization routine. Case I results for GA and *fmincon* are shown in Table 4.1. Figure 4.1 shows the target and traced points in X-Y (Case I with GA). Since *fmincon* yields only a single result, it is only included in Table 4.1 as a separate column. Objective functions are the same with GA, they are not rewritten here. Related mechanism is drawn in next section with Excel.

Table 4.2 presents target and traced point with GA. To prepare Table 4.2, the following procedure is applied with help of Matlab Optimization toolbox (GA) and Excel spread sheet.

(i) The main program written previously called four_bar.m is called and chosen target points are typed with initial mechanism parameters.

(ii) Matlab Optimization Toolbox is then activated and operation values for optimization are typed, the program is run afterwards. These details are included in Chapter 3.

(iii) The program at the end gives optimum values of the mechanism as $r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_2$. By using these values, the coupler positions are calculated using equations (4.10a) and (4.10b) successively.

(iv) Then Excel program is activated to perform kinematic analysis with optimum values and also to draw the mechanism of interest. Mechanism is presented in next section.

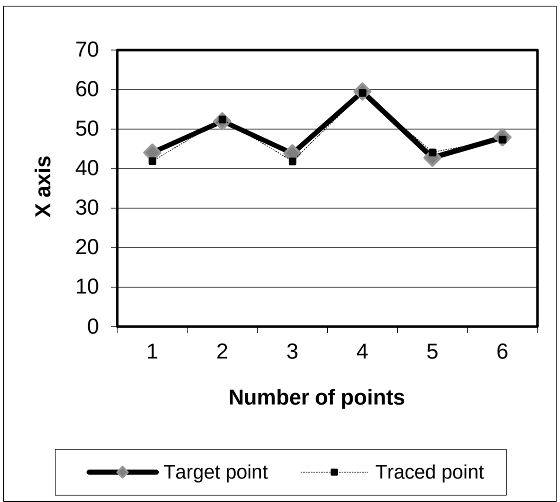
(v) By using this mechanism, traced points are calculated for Case I in Table 4.2.

Table 4.1. Results for **GA** and ***(fmincon)*-Case I**

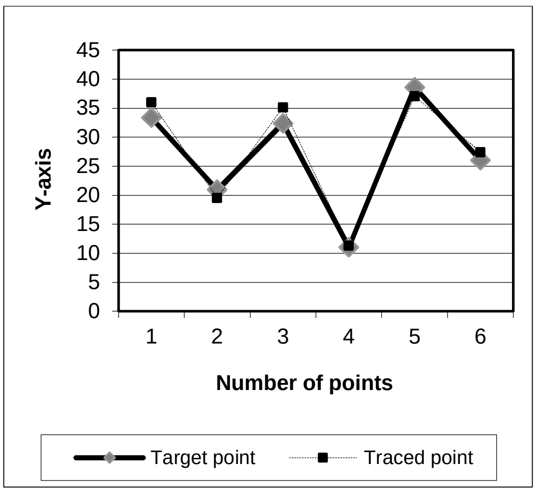
	[20,20]	**[20,25]**	**[20,30]**	**[20,35]**	**[20,40]**	**[20,45]**	***fmincon***
r_1	56.338	59.976	48.013	52.64	58.907	54.348	40.00
r_2	54.992	55.015	53.745	59.839	57.403	54.013	50.00
r_3	55.369	64.897	53.878	50.622	52.068	52.201	50.00
r_4	54.009	59.879	59.593	57.828	50.566	51.846	60.00
r_{cx}	0.626	0.694	0.332	0.653	0.113	0.238	32.00
r_{cy}	0.306	0.338	0.823	0.692	0.206	0.669	0.00
Θ_2	0.652	0.394	0.524	0.184	0.746	0.498	0.524
f_{obj}	198.1	107.419	66.73	76.056	135.37	244.69	

Table 4.2. Target and traced points (***Case I-GA***)

POINTS	***TARGET-X***	***TARGET-Y***	***TRACED-X***	***TRACED-Y***
(20,20)	44.011	33.351	41.874	35.997
(20,25)	51.965	20.921	52.404	19.529
(20,30)	43.839	32.381	41.845	35.122
(20,35)	59.472	11.041	59.169	11.311
(20,40)	42.753	38.602	44.131	37.070
(20,45)	47.869	25.997	47.368	27.398



(a)



(b)

Figure 4.2. Target and traced points in X-Y with precision points (Case I-GA)

4.4.2. Case II- Path generation without prescribed timing

There are six coupler points required to find out an optimal solution at the end. These points are designed to trace a horizontal straight line by changing X coordinate only. The problem is then defined by;

(i) The design variables are;

$X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_2]$ Where *i=1.2…..N and N=6*

(ii) Target points are chosen as:

$[C_{xT}^i, C_{yT}^i] = [(20,20),(25,20),(30,20),(35,20),(40,20),(45,20)]$

(iii) Limits of the variables;

$r_1, r_2, r_3, r_4 \in [13,70]$. $r_{cx}, r_{cy} \in [-60,60]$ and $\theta_2 \in [0,2\pi]$ where *i=1.2…..N and N=6*

(iv) Parameters of GA;

Population Size (PS) =20. Crossover Possibility (CP) =0.8.

Mutation Possibility (MP)-uniform=0.1. Selection type=roulette

(v) fmincon conditions;

Maximum iterations=400

Optimization Toolbox command *fmincon* is compared with GA here. The program is run by following steps given in Chapter 3. m-files are written for four bar mechanism (see Appendix). Case II results are shown in Table 4.3 for GA and *fmincon*. Here *fmincon* is presented as a separate column in Table 4.3. Table 4.4 presents target and traced point with GA. Here all procedure is the same given in Case I. These points are calculated using equations (4.10a) and (4.10b).

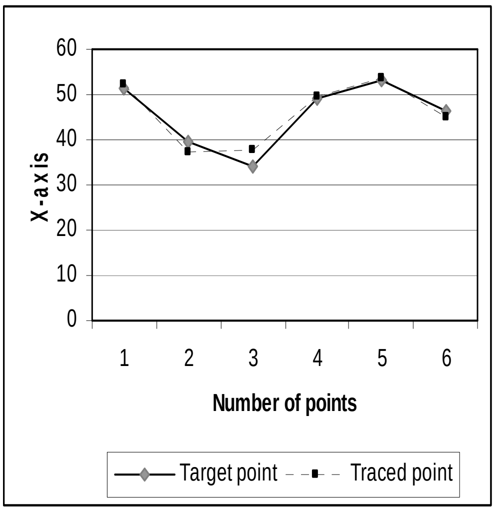
Figure 4.3 shows target and traced points in X-Y for Case I applying GA. Mechanism referring this case will be given in next section.

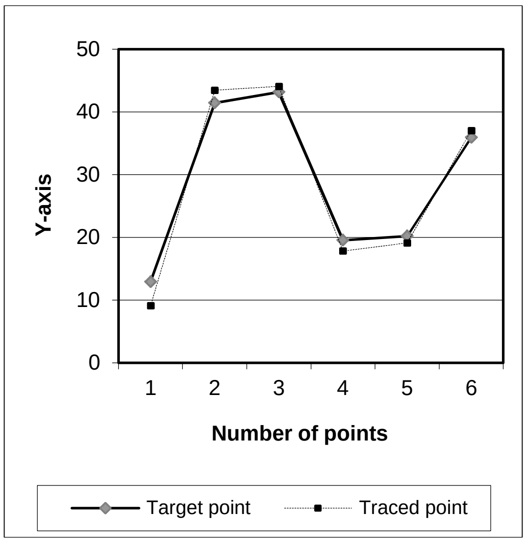
Table 4.3. Results for **GA** and ***(fmincon*)-Case II**

	[20,20]	**[25,20]**	**[30,20]**	**[35,20]**	**[40,20]**	**[45,20]**	***fmincon***
r_1	58.449	59.842	55.556	59.710	59.982	59.56	40
r_2	52.168	56.676	57.510	52.029	55.861	57.645	50
r_3	63.483	62.693	57.189	67.034	63.332	54.512	50
r_4	57.206	59.525	59.142	59.351	59.207	52.599	60
r_{cx}	0.752	0.816	0.228	0.759	0.708	0.923	32
r_{cy}	0.750	0.008	0.673	0.086	0.683	0.526	0.0
Θ_2	0.235	0.833	0.942	0.395	0.371	0.664	0.524
f_{obj}	198.1	303.00	458.440	663.623	918.797	1223	

Table 4.4. Target and traced points (*Case II-GA*)

POINTS	***TARGET-X***	***TARGET-Y***	***TRACED-X***	***TRACED-Y***
(20,20)	51.4	12.91	52.473	9.08
(25,20)	39.41	41.45	37.245	43.455
(30,20)	33.99	43.19	37.676	44.068
(35,20)	48.98	19.55	49.653	17.828
(40,20)	53.166	20.23	53.747	19.128
(45,20)	46.32	35.946	45.22	37.036





(a) (b)

Figure 4.3. Target and traced points in X-Y with precision points (Case II-GA)

4.4.3. Case III- Path generation without prescribed timing

Number of precision points is increased to nine. There are nine coupler points required to find out an optimal solution at the end. These points are designed to trace by changing X and Y coordinates. The problem is then defined by;

(i) The design variables are;

$X = [r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_2]$ Where *i=1.2.….N and N=6*

(ii) Target points are chosen as:

$[C_{xT}^i, C_{yT}^i] = [(10{,}5),(15{,}20),(5{,}20),(20{,}10),(25{,}15),(15{,}25),(5{,}10),(10{,}15),(25{,}20)]$

(iii) Limits of the variables;

$r_1, r_2, r_3, r_4 \in [13{,}70]$. $r_{cx}, r_{cy} \in [-60{,}60]$ and $\theta_2 \in [0{,}2\pi]$

where *i=1.2.….N and N=6*

(iv) Parameters of GA;

Population Size (PS) =20. Crossover Possibility (CP) =0.8.

Mutation Possibility (MP)-uniform=0.1. Selection type=Roulette

(v) fmincon conditions;

Maximum iterations=400

Optimization Toolbox command *fmincon is* compared with GA here. The program is run by following steps given in Chapter 3. m-files are written for four bar mechanism (see Appendix). Case III results are shown in Table 4.5 and Table 4.6. Here to prepare Table 4.6, same procedure is applied.

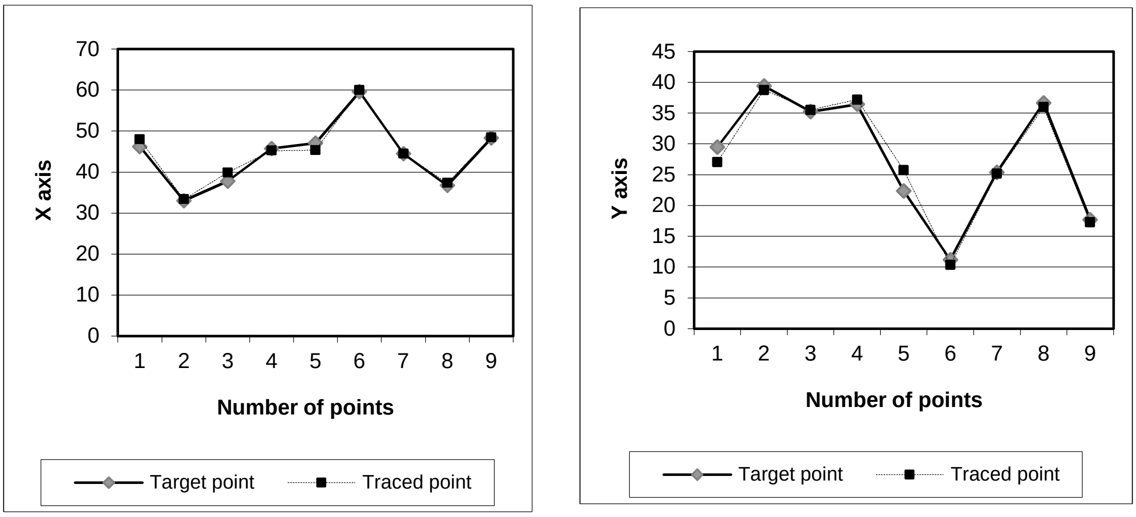
Figure 4.4. presents target and traced points in X-Y with nine points. This mechanism is also drawn using Excel spread sheet in Section 4.5. Calculation of mechanism parameters with *fmincon* is given in Table 4.5 as a separate row. It gives same results with the same objective function with GA.

POINTS	**[10, 5]**	**[15,20]**	**[5, 20]**	**[20,10]**	**[25,15]**	**[15,25]**	**[5, 10]**	**[10,15]**	**[25,20]**	***fmincon***
r_1	55.847	59.056	57.788	57.350	47.687	59.993	41.65	57.828	57.563	40.096
r_2	53.977	50.490	50.537	57.916	51.384	59.69	50.241	51.764	50.387	50.060
r_3	61.648	66.765	58.933	50.673	52.387	50.744	50.737	64.148	62.443	50.01
r_4	59.779	58.198	51.677	51.241	56.084	50.434	59.327	58.081	55.267	59.974
r_{cx}	0.859	0.547	0.824	0.119	0.501	0.857	0.942	0.095	0.964	31.986
r_{cy}	0.898	0.684	0.829	0.826	0.688	0.871	0.215	0.06	0.431	0.00
Θ_2	0.56	0.603	0.755	0.671	0.429	0.167	0.549	0.777	0.331	1.077
f_{obj}	709.797	142.946	182.758	529.464	443.956	52.245	513.941	278.434	875.319	

Table 4.5. Results for GA and *fmincon* –Case III

Table 4.6. Target and traced points (***Case III-GA***)

POINTS	*TARGET-X*	*TARGET-Y*	*TRACED-X*	*TRACED-Y*
[10, 5]	46.210	29.472	47.986	27.035
[15,20]	32.990	39.380	33.327	38.733
[5, 20]	37.771	35.240	39.881	35.507
[20,10]	45.715	36.430	45.199	37.189
[25,15]	47.060	22.370	45.350	25.721
[15,25]	59.570	11.182	60.003	10.361
[5, 10]	44.459	25.377	44.473	25.189
[10,15]	36.694	36.660	37.347	35.964
[25,20]	48.280	17.670	48.403	17.269



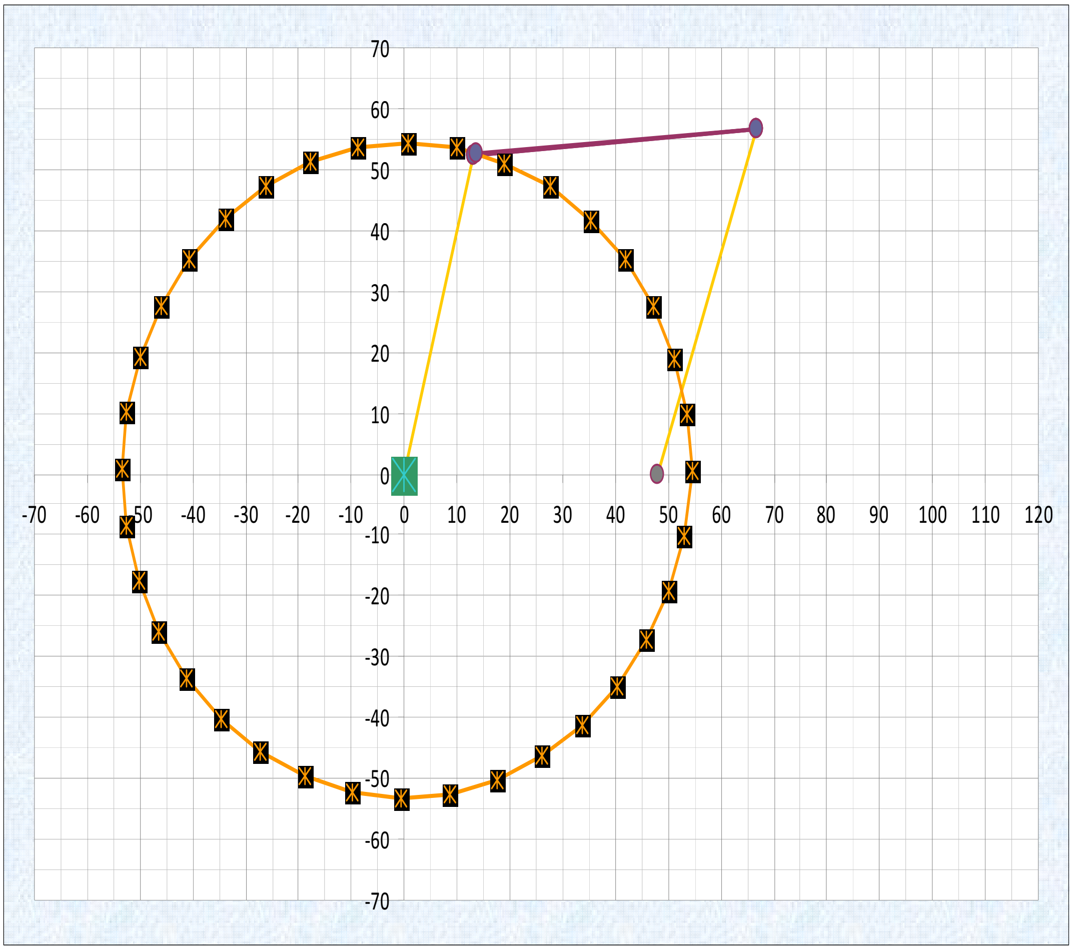
(a) (b)

Figure 4.4. Target and traced points (Case III-GA)

4.5. Studying Mechanisms with Excel Spread Sheet

With the advances of computer technology, mechanism analysis and synthesis using computers is continually developing process. These programs are well written programs and the students can learn on how to define an input and then how to interpret the output. All spreadsheet programs are arranged cells as rows and columns; this depends on the requirement given by the user. Here optimizations results are taken and drawn on a spread sheet Freudenstein's equations are utilized for the synthesis. Initial crank angles are changed successively; different solutions are found and drawn with the mechanism. It is possible to draw coupler curves and its coordinates with velocity and acceleration as well. Then they can be seen on the screen in animated sense. Some study is needed to draw mechanism in Excel. In this study. a previously prepared four bar mechanism code has been applied. [3. 37]

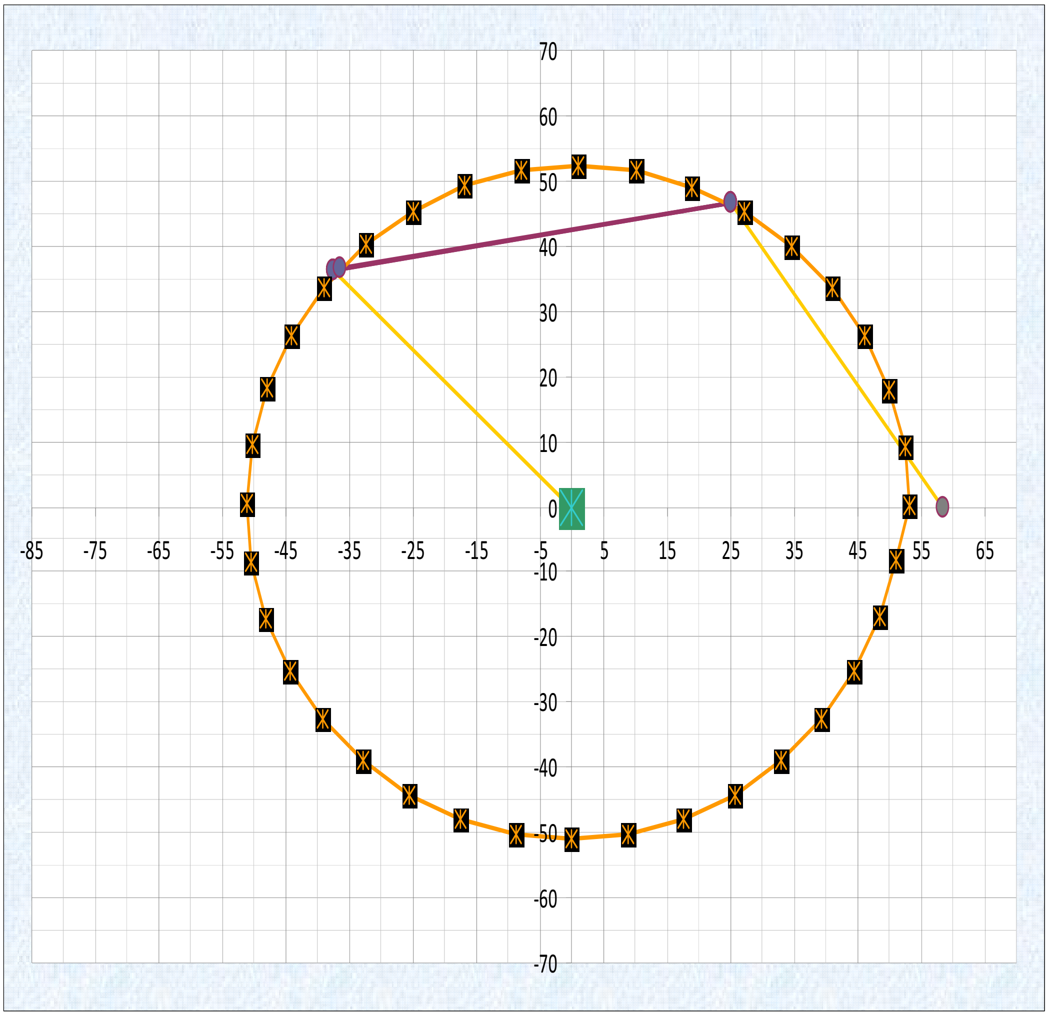
Figure 4.5 shows four bar mechanism for case I. It is possible to get complete behavior of the mechanism by changing in put angle. Referring to Figure 4.1 inputs are given as r_1. r_2. r_3. r_4. r_{cx}. r_{cy}. θ_2 found from optimization. The mechanism is drawn next. If required complete kinematic analysis can be seen as positions. velocities and accelerations for each point separately as well.



r_1=48.013 r_2=53.745 r_3=53.878 r_4=59.593 r_{cx}=0.332 r_{cy}=0.823

Figure 4.5. Four Bar Mechanism -(Case I)

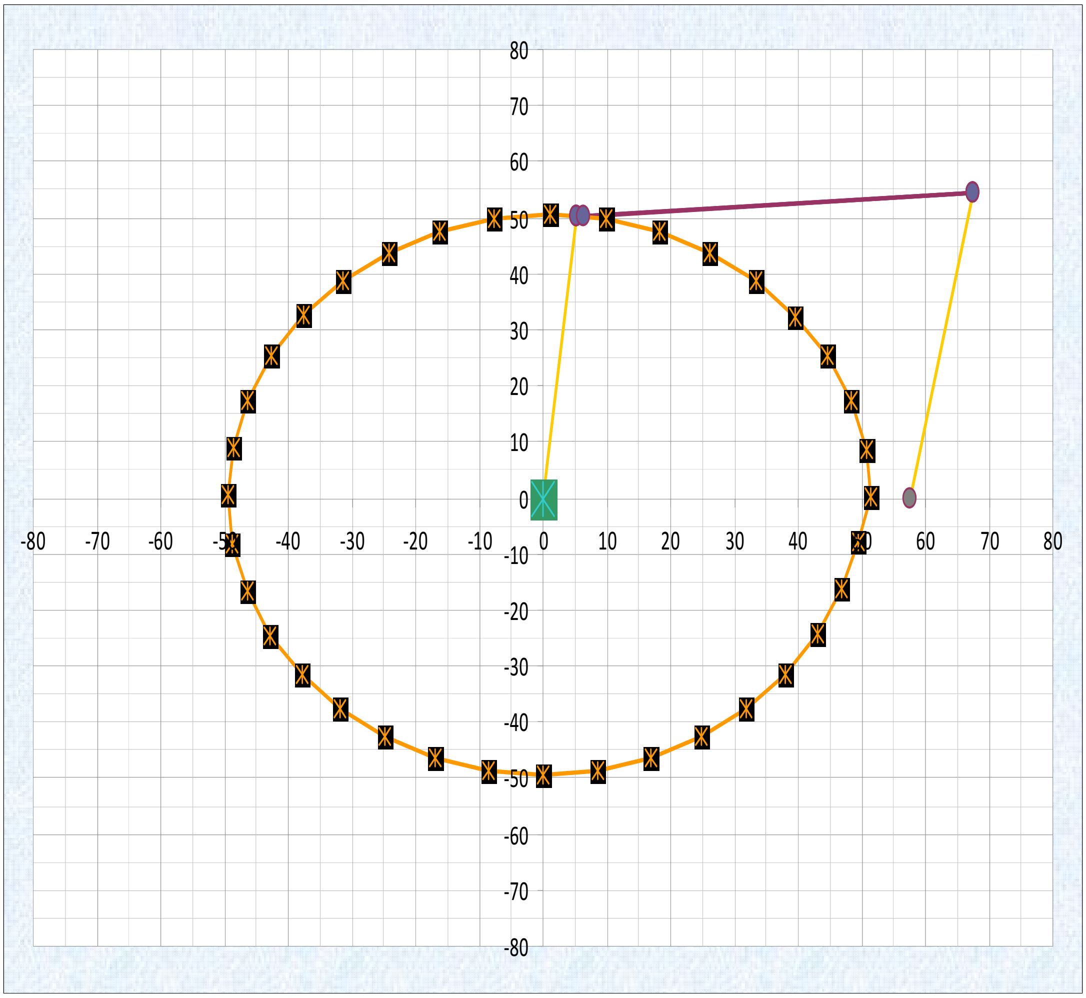
Figure 4.6 shows four bar mechanism for Case II. Similar procedure is applied. With given input parameters found from optimization. mechanism is drawn. While drawing possible mechanisms. Grashof's condition is checked also.



r_1=55.556 r_2=57.510 r_3=57.189 r_4=59.142 r_{cx}=0.228 r_{cy}=0.673

Figure 4.6. Four Bar Mechanism (Case II)

Figure 4.7 shows four bar mechanism for Case III. Mechanism parameters are taken from Table 4.5. the required points are checked using Table 4.6.



r_1=57.563 r_2=50.387 r_3=62.443 r_4=55.267 r_{cx}=0.964 r_{cy}=0.431

Figure 4.7. Four Bar Mechanism (Case III)

4.6. Comparison of Optimization Toolbox (*fmincon*) and GA Optimization

Matlab is an important technical programming which involves many different toolboxes. They can be activated by a simple command which makes the package superior to the others. In this study, the commands used in Matlab Optimization Toolboxes are used and compared with GA Toolbox in optimization. Both algorithms are utilized together to see differences in results in constrained numerical optimization.

Initially some functions are used in '*fmincon*' and GA to see what happens. Having understood the technical coding of the program, it has been used for a four bar mechanism same limit values and constraints are given for both to make a fair comparison. Matlab Optimization Toolbox utilizes different algorithms especially *fmincon* for nonlinear multi dimensional constrained numerical optimization.

Three case studies representing different precision points are presented as Case I-II-III. Six to nine points are used during optimization. Since the starting points and initial values are the same, similar results are seen at the end of optimization. GA shows converging behavior in all calculations.

CHAPTER 5
CONCLUSIONS

5.1. Conclusions

This thesis has presented a study for synthesis of planar mechanisms; specifically on a one degree of freedom planar mechanism. The algorithm is developed only for a well known Grashof's type four bar mechanism. The idea is applicable to all types of planar mechanisms. The only difference will be kinematics analysis of the mechanisms and related constraints. The main advantage seen during implementation is that of its simplicity. Utilization of Optimization Toolbox is performed and a fast convergence to optimal solution is observed. Since the routine is performed directly, there will be no need for superior knowledge during optimization.

The study is basically performed in six parts;

(i) A literature survey is performed on synthesis of planar mechanisms and use of evolutionary algorithms in this subject

(ii) A one dof planar mechanism; specifically four bar mechanism (Grashof type) is studied. Optimized results are obtained for a double crank mechanism as a result of the constraint imposed.

(iii) Basic optimization algorithms are studied. Multi objective nonlinear constraint optimization is decided and applied for planar mechanism problems.

(iv) Evolutionary Algorithms (EAs) are searched in general, but specifically GA is focused on, studied and applied.

(v) Matlab Toolboxes are studied (GA Toolbox, Optimization Toolbox)

(vi) Different case studies are presented for a four bar mechanism. The results are compared with numerical optimization techniques and evolutionary algorithm, GA.

(vii) Mechanisms are drawn using Microsoft Excel, computer spread sheets are prepared to see how the mechanism works with optimum parameters.

5.2. Discussions on Optimization Study

Genetic Algorithm is proven to be an efficient tool for solving synthesis problems. Genetic Algorithm (GA) can handle optimization problems which are nonlinear in character. GA imitates the natural evolution process. Larger populations find the optimal in fewer generations, but larger populations taken longer amount of time to compute. So optimum synthesis is performed both GA based optimization and Matlab command function *fmincon* with the same constraints. Nonlinear constrained optimization is certainly performed.

During standard optimization *fmincon* has given only one single result for different tracing points. This is simply resulted from algorithm itself. When GA based optimization is performed, different points are found during the study. The performance of GA depends on the algorithm parameters like population, crossover, mutation, stopping criteria. There can be different alternative solutions depending on the parameters change in the algorithm. This is not searched in this study. By referring to previous studies, parameters are chosen.

5.3. Recommendations for Further Study

Synthesis study seen during the research has offered promising results. Mechanisms are drawn in Excel spread sheet to show how it works on. It is also possible to perform kinematic analysis while it is operating.

- One may want to use a different EC’s like PSO, ACO or any hybrid algorithm to perform nonlinear constrained optimization with better performances.
- Other optimization algorithms can be used with Matlab. (Toolbox based)

- The constraints imposed in the mechanism may be increased. So working platform on synthesis can be more specific.
- The study may be extended to mechanisms with multi degree of freedom with also inclusion of additional constraints. Alternative mechanisms can be decided.
- The study is specifically applied to four bar mechanism. By studying kinematics of a different type of single degree of mechanism, optimization can be carried out for another type planar mechanism; like slider crank.
- When solving problems in practice on any planar mechanism, numbers of design variables have to be changed simultaneously.

REFERENCES

[1] H. Lipson, '*Evolutionary Synthesis of Kinematic Mechanisms*.', Comp. Synt. Lab., Sibley School of Mech. and Aerospace Eng., and Fac. of Computing and Inf. Science, Comell University, Ithaca, NY-USA.

[2] J.H.Bickford '*Mechanisms for Intermittent Motion*.', Director of production planning, Veeder-Root Company. 'Int. Press Inc., 200 Medison Avenue, N.Y. 10016.

[3] E.Söylemez, '*Mechanisms'*, M.E.T.U Mechanical Engineering Department, Publication Number: 64, Ankara.

[4] A.H.Soni, '*Mechanism Synthesis and Analysis*.', School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma,

[5] R.N.Norton (1992), '*Design of Machinery-An Introduction to the Synthesis and Analysis of Mechanisms and Machines*',McGraw Hill.

[6] R.S.Hartenberg (1964), D.J., '*Kinematic Synthesis of Linkages*, McGraw Hill.

[7] Y.Lui & J. McPhee, '*Automated Kinematik Synthesis of Planar Mechanisms with Revolute Joints.'*, System Design Engineering, University of Waterloo, Ontario, Canada.

[8] Ting-Yu Chen, Chen-Ming Yang (2004), '*Multidisciplinary design optimization of Mechanisms.*', Dept. of Mech. Eng., National Chung Hsing University, Taiwan.

[9] T.Back, D.B.Fogel, T. Michalewicz (2000), '*Evolutionary Computation 1- Basic Algorithms and Operators'*, Taylor and Francis..

[10] A.E.Eiben, J.E. Smith (2003), ‘*Introduction to Evolutionary Computing’*, Springer.

[11] R.L. Haupt, S.E. Haupt (2004), ‘*Practical Genetic Algorithms’*, 2nd Edition, Wiley.

[12] J.C.Hoskins, Glenn A. Kramer (1993), ‘Synthesis of Mechanical Linkages using Artificial Neural Networks and Optimization’, *IEEE*, 822J-822N.

[13] M.H.F. Dado (1996), ‘An automated procedure for dimensional synthesis of planar mechanisms’, *J.King Saud Unv*., Vol.8, Eng. Sci(1), 17-41.

[14] R.C.Blackett (2001), *’Optimal Synthesis of Planar Five Link Mechanisms for the Production of Nonlinear Mechanical Advantage*’, Ms Thesis, Virginia Polytechnic Institute.

[15] P.S.Shiokolas, V. Chandra, J.Kebrle, D. Wilhite (2002), *’Analysis and Simulation for Education using Matlab via the World Wide Web., II Representative Examples- System Simulation and Planar Mechanism Synthesis and Analysis*, , Wiley Periodicals[http:/zodhia.uta.edu/development/,109-121.

[16] J.A Cabrera, A Simon, M. Prado (2002), ‘Optimal Synthesis of Mechanisms with Genetic Algorithms’, *Mechanism and Machine Theory*, 37, 1165-1177.

[17] R.R.Bulatovic, S.R. Djordjevic (2004), ‘Optimal Synthesis of a Four Bar Linkage by Method of Controlled Deviation’, Theoretical Appl. Mech., Vol.31, No:3-4, 264-280.

[18] M.A.Laribi, A.Mlika, L.Romdhane, S.Zeghloul (2004), ‘A Combined Genetic Algorithm-Fuzzy Logic method (GA-FL) in Mechanisms Synthesis’, Mechanism and Machine Theory, 39, 717-735.

[19] K.G.Cheetancheri, Harry H. Cheng (2005), ‘*Computer-Aided Analysis of Mechanisms Using Ch Excel*’, Department of Mech. And Aeronautical Engineering. .

[20] J.F. Collard, P.Fisette, P. Duysinx (2005), ‘Optimal Synthesis of Mechanisms using time varying dimensions and natural coordinates’, *6th World Congress of Structured and Multidisiplinary Optimization*, Rio de Janerio-Brazil,1-10 .

[21] H.H.Cheng, Dung T. Trang (2006),’ Web-Based Interactive Analysis and Animation of Mechanisms’, Transactions of the ASME, Journal of Computing and Information Science in Engineering, Vol.6, 84-90 [http://www.softintegration/webservices/mechanism/]

[22] J.Xie, Y. Chen (2007), ‘*Application Back Propagation Neural Network to Synthesis of Whole Cycle Motion Generation Mechanism’*, 12th IFToMM World Congress-Besancon-France, June 18-21.

[23] Y.Liu, J.McPhee (2007),’Automated Kinematic Synthesis of Planar Mechanisms with Revolute Joints’, *Mechanics Based Design of Structures and Machines,* 35, 405-445.

[24] S.Erkaya, İ.Uzmay (2008), ‘A Neural-Genetic (NN-GA) approach for optimizing mechanisms having joints with clearance’, *Multibody System Dynamics,* 20, 69-83.

[25] N.N. Zadeh, M. Felezi, A. Jamali, M. Ganji (2009), ‘Pareto optimal Synthesis of four-bar mechanisms for path generation’, *Mechanism and Machine Theory*, 22, 180-191.

[26] S.K. Archaryya, M. Mandal (2009), ‘Performance of EAs for four-bar linkage synthesis’, *Mechanism and Machine Theory*, 44, 1784-1794.

[27] A. Kentli, A.K. Kar, E.Taçgın (2009), ‘*Genetic Coding Application to Synthesis of Planar Mechanisms*’, 5th. Int. Adv. Tech. Symposium (IATS’09), May 13-15, Karabük-Türkiye.

[28] K. Sedlaczek, P.Eberhard, ‘Augmented Lagrangian Particle Swarm Optimization in Mechanism Design’, *Journal of System Design and Dynamics*, 410-421. [www.itm.uni-stuttgart.de{research/alpso]

[29] F.Penunuri, R.P. Escalante, C. Villanueva, D. Pech-Oy (2011), ‘Optimum Synthesis of Mechanism for single and hybrid tasks using differential evolution’, *Mechanism and Machine Theory*, Vol.46, No.10, 1335-1349.

[30] Tokuz L.C(1992), ‘*Hybrid Machine Modelling and Control’*, Ph.D. Thesis, Liverpool Polytechnic.

[31] Greenough J.D., Bradshow W.K, Gilmartin M.J., Jones J.R (1995)., *‘Design of Hybrid Machines’*, Proc. 9th World Congress on the Theory of Machines and Mechanisms, Milan-Italy, V.4, 2501-2505.

[32] Connor M.A, Douglas S.S., Gilmartin M.J (1995)., ‘The Synthesis of Hybrid Five Bar Path Generating Mechanisms Using Genetic algorithms’, *Genetic Algorithms in Engineering Systems: Innovations and Applications*, 313-318, IEE .

[33] Zhou H., Edmund Cheung H.M (2000), ’Analysis and Optimal Synthesis of Hybrid Five Bar Linkages’, *Mechatronics*, 11, 283-300.

[34] Kirecci A., Dülger L.C.(2000), ‘A study on a hybrid actuator’, *Mechanism and Machine Theory*, Vol.35, No.8, 1141-1149.

[35] Kirecci A., Dülger L.C. (2003), ‘Motion design and implementation for a hybrid drive system’, *IMechE-Vol.217, Part I, J. Systems and Control Engineering*, 299-302.

[36] Zhang Ke (2008), ‘*Synthesis of a Hybrid Five Bar Mechanism with Particle Swarm Optimization Algorithm’*, Part I, LNCS5262, Springer-Verger, Berlin Heidelberg, 873-882.

[37] Eres Söylemez (2009), 'Using Computer Spread Sheets in Teaching Mechanisms', Springer- Proceedings of EUCOMES 08, 45-53.

[38] J.W.Chinneck (2000), '*Practical Optimization: A Gentle Introduction'*. (http:www.sce.carleton.ca/faculty/chinneck/po.html)

[39] *Optimization Toolbox User's Guide* (2001)-(For use with Matlab)

[40] A Geletu (2007), '*Solving Optimization Problems using the Matlab-Optimization Toolbox: a Tutorial',* TU Ilmenau, Faculty of Mathematics and Natural Sciences.

[41] A. Chipperfield, P.Fleming, H.Pohlheim, C. Fonseca, '*Genetic Algorithm Toolbox (For Use with MATLAB, Version 1.2)*-User's Guide.

[42] '*Genetic Algorithm and Direct Search Toolbox* (For Use with MATLAB, Version 1.2)-User's Guide.

[43] Global Optimization Toolbox 3, User's Guide, Matlab-Mathworks

[44] Hartmut Pohlheim (2006), *GEATbx: Introduction: Evolutionary Algorithms, Overview, Methods and Operators,* Version 3.8.

APPENDIX A

MATLAB FOUR BAR CODE

FITNESS FUNCTION

```
function f=four_bar(r1,r2,r3,r4,rcx,rcy,xo,yo,th0,th2);
%x=[r1,r2,r3,r4,rcx,rcy,xo,y0,th0,th2];
r1=13;
r2=50;
r3=64;
r4=50;
rcx=r3/2;
rcy=0;
th2=pi/6;
th0=0/pi;
xo=0;
yo=0;


k1=r1/r2;
k2=r1/r4;
k3=(r2^2-r3^2+r4^2+r1^2)/2*r2*r4;
k4=r1/r3;
k5 = (r4^2-r1^2-r2^2-r3^2)/(2*r2*r3);

a=cos(th2)-k1-k2*cos(th2)+k3;
b=-2*sin(th2);
c=k1-(k2+1)*cos(th2)+k3;
d=cos(th2)-k1+k4*cos(th2)+k5;
e=-2*sin(th2);
f=k1+(k4-1)*cos(th2)+k5;

th4=2*atan((-b+sqrt(b^2-4*a*c))/(2*a));
th3=2*atan((-e+sqrt(e^2-4*d*f))/(2*d));

cxr=r2*cos(th2)+rcx*cos(th3)-rcy*sin(th3);
cyr=r2*sin(th2)+rcx*sin(th3)-rcy*cos(th3);

mat2=[cos(th0),-sin(th0);sin(th0),cos(th0)];
mat3=[cxr;cyr];
mat4=[xo;yo];
%[cx;cy]=[cos(th0),-
sin(th0);sin(th0),cos(th0)]*[cxr;cyr]+[x0;y0]

mat1=mat2*mat3+mat4;
```

```
cx=mat1(1);
cy=mat1(2);
cxd=20;
cyd=45;
%Cd=[cxd;cyd]';
%C=[cx;cy]';

%mat1=[cx;cy];
%r1=r2*cos(teta2)+r3*cos(teta3)-r4*cos(teta4);
%r4=(r2*sin(teta2)+ r3*sin(teta3)/sin(teta4));

%k1*cos(teta4)-k2*cos(teta2)+k3=cos(teta2-teta4);
%k1*cos(teta3)-k4*cos(teta2)+k5=cos(teta2-teta3);

f=(cxd-cx)^2+(cyd-cy)^2;
```

Writing up the constraint function

```
function[c, ceq] = fourbarcon(x)
%nonlinear inequality constraints

c= [];
%nonlinear equality constraints
ceq = [x(2)+x(3)-x(4)-x(1)];
```