MISSILE TRAJECTORY OPTIMIZATION USING GENETIC ALGORITHM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MELTEM SOYLUOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF AERONAUTICAL ENGINEERING

DECEMBER 2001

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Tayfur ÖZTÜRK
Director

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

Prof. Dr. Nafiz ALEMDAROĞLU
Head of the 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.

Assoc. Prof. Dr. Ozan TEKİNALP
Supervisor

Examining Committee Members

Prof. Dr. Nafiz ALEMDAROĞLU

Prof. Dr. Cevdet ÇELENLİGİL

Assoc. Prof. Dr. Mehmet AKGÜN

Assoc. Prof. Dr. Ozan TEKİNALP

Volkan NALBANTOĞLU (Ph.D.)

ABSTRACT

MISSILE TRAJECTORY OPTIMIZATION USING GENETIC ALGORITHM

Soyluoğlu, Meltem

M.S., Department of Aeronautical Engineering

Supervisor : Assoc.Prof. Dr. Ozan Tekinalp

December 2001, 119 pages

In this thesis, application of genetic algorithm to missile trajectory optimization problem is investigated. The missile flight mechanics model is a two degrees of freedom trim flight model with proper aerodynamic coefficients interpolated from the appropriate tables generated with missile DATCOM. The equations are parameterized using angle of attack values at the nodes. The nodes are selected to form equal time intervals until burnout. After burnout they are chosen to form either equal intervals in time or equal energy consumption intervals. Both approaches are investigated and compared. Two fundamental problems are addressed: maximum range trajectory optimization problem and specified range minimum flight time problem. In both problems, terminal constraints are imposed and the effectiveness of the method is demonstrated. A proper constraint handling approach is proposed. A parametric study is conducted to evaluate the effect of various algorithm parameters, such as population size, number of genes used (angle of attack values), crossover, and mutation probabilities on trajectory optimization problem. The results of this parametric study are given and discussed.

Keywords: Trajectory Optimization, Genetic Algorithm, Missile Flight Mechanics.

ÖZ

GENETİK ALGORİTMASI KULLANILARAK FÜZELER İÇİN YÖRÜNGE OPTİMİZASYONU

Soyluoğlu, Meltem

Yüksek Lisans, Havacılık Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Ozan Tekinalp

Aralık 2001, 119 sayfa

Bu tezde, genetik algoritmasının füze yörünge optimizasyon problemlerine uygulanması incelenmiştir. Füze uçuş mekaniği modeli olarak iki serbestlik dereceli uçuş modeli kullanılmıştır. Modelin ihtiyaç duyduğu aerodinamik kuvvetler Missile DATCOM programıyla elde edilen katsayılar tablosu kullanılarak hesaplanmıştır. Düğümlerdeki hücüm açıları yörünge eniyilemesi parametreleri olarak seçilmiştir. Düğümler yakıt bitene kadar eşit zaman aralıklarını ifade etmektedir. Yakıt bittikten sonra düğümler için eşit zaman aralıklarında ve eşit enerji tüketim aralıklarını kullanan iki yaklaşım incelenmiş ve sonuçları karşılaştırılmıştır. İki eniyileme problemi çözülmüştür. Bunlar; en uzak menzile giden menzil yörünge eniyilemesi problemi ile, verilmiş bir menzile en kısa uçuş süresinde giden yörünge eniyilemesi problemleridir. Her iki problemde de yörünge sonunda elde edilmesi gereken değerler için bazı kısıtlamalar tanımlanmış ve yöntemin etkinliği sınanmıştır. Kısıtlama değerlerinin sağlanmasına yönelik uygun bir yöntem kullanılmıştır. Genetik algoritmasında kullanılan popülasyon büyüklüğü, gen (hücum açı değerleri) sayısı, çaprazlama ve mutasyon olasılıkları gibi parametrelerin yörünge optimizasyonu problemi üzerindeki etkilerini incelemek amacıyla parametre analiz çalışması yapılmış ve elde edilen sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Yörünge Optimizasyonu, Genetik Algoritması, Füze Uçuş Mekaniği.

To my love, Kaan Akar...

ACKNOWLEDGEMENTS

I express sincere appreciation to my supervisor Assoc. Prof. Dr. Ozan Tekinalp for his guidance and support throughout this study.

Special thanks to my father Cevdet, my mother Güneş, and my sister Mehtap for their love and support at all stages of my life.

And Kaan... This work would not be possible without your support, patience and understanding.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

NOMENCLATURE

C_A	Axial force coefficient
C_D	Drag coefficient
C_L	Lift coefficient
C_N	Normal force coefficient
C_m	Pitching moment coefficient
c	Speed of sound
f*	Global optimum
h	Altitude
k_i	Penalty coefficients
M	Mach number
m	Missile mass
m_t	Missile total mass
m_p	Total propellant mass
q	Dynamic pressure
q	Cumulative probability
P	Pressure
p	Probability of selection
R	Universal gas constant

r	Range
S_{ref}	Aerodynamic reference area
V	Total velocity
V_s	Speed of sound
t	Simulation time
t_b	Burnout time
T	Atmospheric temperature
X_{cg}	Center of gravity location
X_{cp}	Center of pressure location
X_{LE}	Wing leading edge location
α	Angle of attack
γ	Flight path angle, Specific heat ratio
γ_{air}	Specific heat ratio of air
θ	Pitch angle for missile
ρ	Atmospheric density
Λ	Sweep angle
ASM	Air-to-surface missile
DOF	Degree of freedom
FP	Floating Point
GA	Genetic algorithm
INS	Inertial navigation system
SA	Simulated annealing

CHAPTER 1

INTRODUCTION

1.1 Purpose

Design process of a missile comprises determinative decisions involving trade-offs among the proposed goals. The success of those decisions is based on the proper modeling of the missile and the estimation of the effects of various parameters on the optimal behavior of the designed system. Various optimization techniques are currently being used in airborne systems designs in order to obtain optimal solutions for the desired goals subject to the constraints of the particular mission. Among those techniques, "Genetic Algorithm" has raised accelerated interest through the last decade due to its success in finding the global optima against relatively larger numbers of constraints.

The purpose of this work is to investigate the applicability of genetic heuristics to missile trajectory optimization.

1.2 Guided Air-to-Surface Missiles

Guided Missiles are self-propelled aerial projectiles, guided in flight toward a target either automatically or remotely. Guided missiles vary widely in size and type, ranging from large strategic ballistic missiles with nuclear warheads to small, portable rockets carried by foot soldiers. Although most are military weapons with explosive warheads, others may carry scientific instruments for gathering information within or above the earth's atmosphere.

Before World War II guided missiles were limited to experimental, pilotless aircraft controlled by radio. During the war, however, rapid technological advances in such fields as aerodynamics, electronics, jet and rocket propulsion, radar, servomechanisms, inertial guidance and control systems, and aircraft structures, coupled with the intensive search for better weapons, led to the construction, testing, and finally mass production of the modern guided missile. The Persian Gulf War of 1991 marked the first time that precision-guided missiles played an important role in warfare. A coalition led by the United States launched several kinds of guided missiles against Iraq.

Guided missiles today are grouped into four launch-to-target categories: surface-to-surface, surface-to-air, air-to-surface, and air-to-air. Among them air-to-surface missiles (ASM) are increasingly being used.

ASM's are launched from aircraft against targets on the ground or at sea. These missiles are generally short-range, light, rocket-powered projectiles with sophisticated internal guidance systems. During the Persian Gulf War, allied forces destroyed Iraqi radar-controlled missile sites with air-to-surface High Speed Anti-

Radiation Missiles (HARM's), that homes in the enemy radar beam and destroys the transmitter.

1.3 Optimization Algorithms

Gradient-based techniques are commonly used in design optimization. The gradient-based algorithms of optimization are a class of search methods for real-valued functions. These methods use the gradient of a given function as well as function values. They have been extensively employed for nonlinear programming problems [1].

Although, commonly used the gradient-based methods are subject to several undesirable restrictions. First, these methods require knowledge of the aerodynamic derivatives for each parameter, or a combination of parameters for higher order coupling. Such derivatives are not easily determined and must be continually recalculated as the design proceeds; few aerodynamic design problems have linear derivatives, especially in multiple dimensions. Second, gradient-based optimizers must start with a specified set of initial parameters, which can bias future solutions towards a local optimum in the vicinity of the starting point. Also, gradient-based methods cannot be applied to problems with discontinuities in the design space because the derivatives in these regions are not defined. Another restriction of gradient-based algorithms is that, as the number of design variables increases and coupling of the variables occurs, they lose much of their efficiency [2].

Most design problems are characterized by mixed continuous-discrete variables, and discontinuous and non-convex design surfaces. If standard nonlinear programming techniques are used for this type of problems, they will be inefficient and may not converge. Gradient-based methods cannot use discrete variables since gradient with respect to integer numbers are not defined. Non-gradient type algorithms specialize in performing a complete search of the design space and as a consequence are often referred to as global search algorithms.

As an example to the application of gradient-type algorithms Tava and Suzuki [3], solved three trajectory optimization problems using a sequential quadratic programming code, BDH, and compared the results with their known analytical solutions. The results showed a good approximation to the optimal value of the objective functions in all cases.

Arslan and Tekinalp have used direct collocation to trajectory optimization problem together with nonlinear programming. This technique is applied to a trajectory optimization problem of an air-to-surface missile configuration. Several case studies are conducted such as, maximum range, minimum time of flight to a given range, minimum weight to a specified range [4-6].

Two relatively new optimization methods are the genetic algorithm and simulated annealing. The genetic algorithm models natural selection and evolution, while simulated annealing models the annealing process. Both methods generate new points in the search space by applying operators to current points and statistically moving toward more optimal places in the search space. These algorithms do not require the derivatives of the cost function and can thus deal with discrete parameters and noncontinuous cost functions. Both techniques have been successfully applied to numerous problems in a number of different areas [7].

In the early 1980's the method of simulated annealing was introduced by Kirkpatrick and coworkers (1983), based on ideas formulated in the early 1950's (Metropolis et al., 1953). This method simulates the annealing process in which a substance is heated to a temperature above its melting temperature, and then cooled gradually to produce a crystalline lattice, which minimizes its energy probability distribution. If the cooling proceeds too quickly the crystal never forms, and the substance becomes an amorphous mass with a higher than optimum energy state. The key to crystal formulation is carefully controlling the rate of change of temperature. What makes the simulated annealing unique is the addition of the control parameter, analogous to the temperature, which controls the speed of descent of the algorithm into the optimum cost function value. This algorithm has solved the traveling salesman problem and has been applied to a wide variety of problems [8].

Bélisle et al. has developed a simulated annealing (SA) algorithm for continuous optimization, called Hide-and-Seek. The given details and discussion of the algorithm together with some conducted numerical experiments have shown that Hide-and-Seek significantly outperforms two multi-start global optimization schemes in efficiency [9]. Lu and Khan employed the algorithm to solve the nonsmooth trajectory optimization for a high-performance, rigid-body aircraft. The objective of the optimization was to find the optimal control histories for the aircraft to minimize the flight time for a specified maneuver. This was a nonsmooth trajectory optimization problem and indirect trajectory optimization methods based on calculus of variations were not applicable. Hide-and-Seek have consistently performed much better compared to two well-known conventional nongradient algorithms: Principal Axis, which is based on Powell's method, and Nelder-Mead Simplex. The application of this simulated annealing algorithm to trajectory optimization for a realistic, nonlinear, 6-DOF high-performance fighter aircraft demonstrated the superiority of the algorithm [10].

Utalay and Tekinalp used simulated annealing (SA) technique to find the maximum range flight trajectory and specified range minimum flight time trajectory of an air to surface missile (ASM). The specified range minimum weight missile configuration was also found considering both the control parameters and missile engine design parameters such as thrust and burnout time [11-12].

Bingöl and Tekinalp, proposed improvements to the basic Hide and Seek algorithm. They also used the algorithm for multidisciplinary missile design optimization. The design optimization problem included disciplines of flight mechanics, propulsion unit design including structural models, and aerodynamics [13].

In this study, genetic algorithm is used as the optimization technique for trajectory optimization. It is a subset of evolutionary algorithms that model biological processes to optimize highly complex cost functions .The method was developed by John Holland (1975) over the course of the 1960s and 1970s and finally popularized by one of his students, David Goldberg [7].

A Genetic Algorithm has been used by Akira Oyama to optimize a wing shape for a generic subsonic transportation aircraft by using Navier-Stokes computations. It was necessary to overcome the enormous computational time for this optimization. The optimization procedure was computer-intensive, yet only a local optimum was hoped for. To accomplish this task satisfactorily, a global optimization algorithm and a powerful computer facility was required. Among the considered optimization algorithms, Genetic Algorithm was attractive for aerodynamic optimization since it is capable of finding a more global optimum than a simple hill climbing technology. Designed wings showed a trade-off between an increase of the airfoil thickness driven by a structural constraint and a reduction of the wave drag produced by a shock wave. GA has found the best feasible solution in the given design constraints [14].

Anderson has examined the use of genetic algorithms to determine high efficiency missile geometries. He demonstrated the capability of these algorithms to determine highly efficient and robust missile aerodynamic designs, given a variety of design goals and constraints. He proved both the learning capability of genetic algorithms and the power of such algorithms for multi-objective optimization in three case studies. First case study performed for the maximization of normal force coefficient, second case for the high normal force and low axial force and the third case for high normal force and low axial force while minimal static margin is aimed. Results have indicated that the genetic algorithm is capable of designing aerodynamic shapes that perform well in either single or multiple goal applications [2].

Another application of GAs for the purpose of mission planning entails sending spacecraft to rendezvous with comets, landing on their surface and taking samples. A GA was developed to determine a complex set of maneuvers to fly over five candidate landing sites. The objective was to minimize the propellant required by the chemical propulsion system during the fly-over of the five candidate landing sites. The maneuvering strategy is complicated by the existence of multiple nonlinear practical constraints. A gradient-based algorithm previously developed for the purpose of determining fly-over maneuvers was found to be incapable of dealing with the posed problem due to the number of fly-over sites and complicated constraints. The GA, however, was able to provide several unique and highly efficient solutions [15].

The feasibility of using a genetic algorithm for the optimization of an ascent trajectory from the surface of the moon is investigated by Pinon. The genetic algorithm generates a thrust profile for each trajectory with the overall objective of minimizing the flight time for a given terminal orbit. The GA results were compared with an analytical solution and were also measured against the results obtained by a calculus-based optimization method. Both comparisons were favourable. During the study, several local optima were found to exist. A hybrid methodology was developed where the GA is used to determine a nominal trajectory, and the calculus-based method then optimizes the GA derived solution. The hybrid technique outperformed the GA alone. The study concluded that genetic algorithms are a feasible method for trajectory optimization [16].

1.4 Original Contributions

Main contributions of this thesis are:

- Two missile trajectory optimization problems are formulated and solved using genetic algorithm. These are maximum range trajectory optimization problem and specified range minimum flight time problem. For this purpose a hypothetical air to surface missile is used with given launch conditions and specified impact conditions. The results are compared to the results obtained by simulated annealing algorithm [13].

- A parametric study is conducted to investigate the effect of the parameters of the genetic algorithm such as population size, number of genes, cross over and mutation probabilities on the trajectory optimization problem.
- A variable crossover probability in which the crossover probability changes during the optimization process is proposed and tested. It is found that variable crossover gives better results.
- The sensitivity of the trajectory to input angle of attack parameters, is investigated through simulations conducted.

1.5 Scope

The organization of the chapters in this thesis is as follows. In Chapter 2, flight mechanics model of a two-degrees of freedom missile is presented. In Chapter 3, genetic heuristics is introduced and genetic algorithm is described. Then it is applied to a simple trajectory optimization problem with a known analytical solution. In Chapter 4, first, the air to surface missile trajectory optimization problem is formulated to maximize the range for given launch and impact conditions. Second, for a specified range and impact conditions, minimum flight time trajectory optimization problem is presented and discussed. Then a parametric study conducted to investigate the behaviour of a genetic algorithm in missile trajectory optimization problem is presented. In particular following parameters are considered: population size, number of genes, fixed and variable crossover probabilities, mutation probabilities. The effect of initial population on the convergence behaviour is also considered. Finally, the effect of perturbation on the angle of attack inputs to the missile range is discussed through a case study. In Chapter 5, conclusions are given and the future work is stated. In Appendix-A, air to surface missile model is detailed and Missile DATCOM™ program capabilities are explained. In Appendix-B, first, maximum range trajectory optimization results obtained using GA and SA are plotted together. Then, desired range minimum flight time, and maximum range trajectory optimization graphs are presented.

CHAPTER 2

MATHEMATICAL MODELING

The evaluation of a missile behaviour requires the mathematical modelling of its aerodynamic, propulsive and structural properties that may in turn be optimized in order to obtain an optimum design satisfying the mission requirements. It may not be quite easy to find a straight-forward solution to this multi-disciplinary design optimization problem. A common approach is to optimize one or a few of the parameters involved while keeping the others constant at a time. Arslan [4] focused on the trajectory optimization together with the propulsion unit design for a particular missile configuration. Utalay [11] used the Hide-and-Seek optimization method, in order to optimize the trajectory control parameters together with the propulsion unit design in a combined manner for a given missile configuration. Bingöl [13] considered the aerodynamic shape of the missile, together with weight and center of gravity calculations in addition to the propulsion unit and the trajectory control parameters.

In this thesis, the trajectory control parameters of an air to surface missile are optimized to attain a given intercept angle and intercept velocity at the maximum possible range using a genetic algorithm. An initial baseline missile configuration with well-defined characteristics is used as a starting point to the optimization. The mathematical model based on these characteristics is illustrated in the following pages.

2.1 Mathematical Modeling of an Air-to-Surface Missile

In this section, a mathematical model for an ASM is defined including equation of motion, aerodynamics, mass and atmospheric characteristics. The ASM configuration chosen for this thesis is given in Appendix A.

In this study, less effective variables are ignored and most critical ones are chosen to decrease the number of mathematical operations.

2.1.1 Equations of Motion

In this thesis a two degrees of freedom model is used with the following assumptions;

- The 2-DOF model considers the motion in the vertical plane as a function of X and Z with the coordinate system shown in Figure 2.1.

- The missile is in the trim condition, that is the net moment on the missile is zero(C_M=0).

- The equations are based on constant gravity and a flat nonrotating earth.

- The orientation of the missile with the free stream is given by the angle of attack.

According to these assumptions, the 2-DOF equations of motion are as follows:

$$\dot{r} = V\cos\gamma \qquad (2.1)$$

$$\dot{h} = V\sin\gamma \qquad (2.2)$$

$$\dot{V} = \frac{1}{m}(Thrust \cdot \cos\alpha - Drag) - g\sin\gamma \qquad (2.3)$$

$$\dot{\gamma} = \frac{1}{mV}(Thrust \cdot \sin\alpha + Lift) - \frac{g\cos\gamma}{V} \qquad (2.4)$$

$$\theta = \gamma + \alpha \qquad (2.5)$$

Here, V is the total velocity of the missile, γ is the flight path angle, θ pitch angle, α angle of attack, m total mass of the missile, r is the range, h is the height, and g is the gravitational acceleration [18]. It is assumed that there is no thrust misalignment and instead of fin deflection, α is the input to the missile.

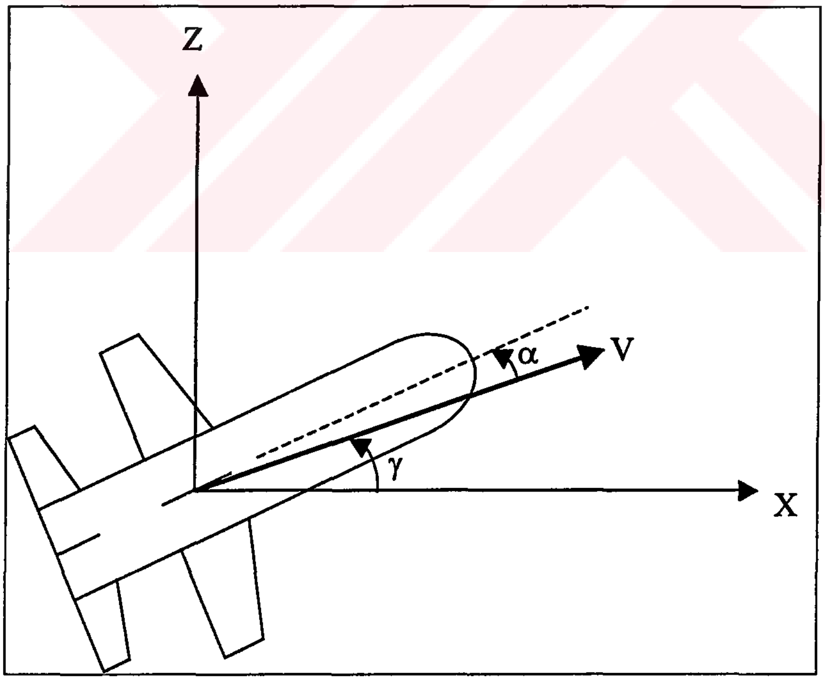


Figure 2.1 2-DOF Coordinate System

2.1.2 Aerodynamics Model

Lift is the aerodynamic force component perpendicular to the total velocity vector of the missile and drag is the one in the direction of the total velocity vector which are calculated by the equations;

$$Drag = q \cdot S_{ref} \cdot C_D \qquad (2.6)$$

$$Lift = q \cdot S_{ref} \cdot C_L \qquad (2.7)$$

where, S_{ref} is the reference cross-sectional area of the missile and the dynamic pressure, q, is given by the equation,

$$q = \frac{1}{2} \cdot \rho \cdot V^2 \qquad (2.8)$$

In supersonic missile-design studies it is more convenient to consider normal forces. The reason for this is that the component section and aerodynamic lifting surfaces are generally symmetrical about the longitudinal axis or chordwise center line; the resultant aerodynamic pressure forces on these symmetrical sections are thus normal to the longitudinal axis or wing chord. In addition, most wind-tunnel strain-gauge balance systems are arranged to measure forces normal and parallel to the longitudinal axis of the model [19]. From Figure 2.2 following force relations are derived:

$$D = A\cos\alpha + N\sin\alpha \qquad (2.9)$$

$$L = N\cos\alpha - A\sin\alpha \qquad (2.10)$$

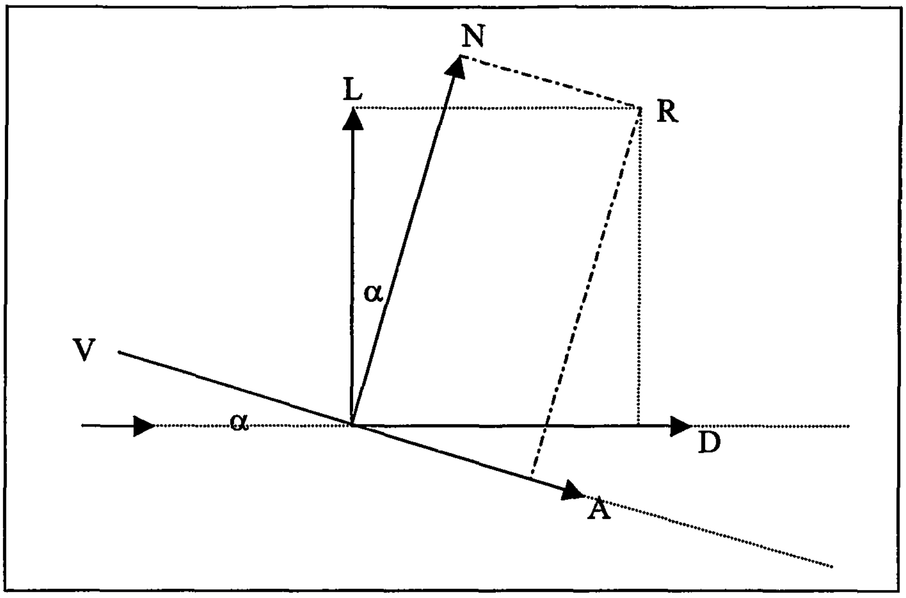


Figure 2.2 Normal Forces

To reduce into coefficient form, divide by $q.S_{ref}$ and Eq. 2.11 & 2.12 are obtained;

$$C_D = C_A \cos\alpha + C_N \sin\alpha \quad (2.11)$$

$$C_L = C_N \cos\alpha - C_A \sin\alpha \quad (2.12)$$

C_N and C_A are both obtained from the outputs of Missile DATCOM™ given in Appendix-A. For a missile in trim condition with a given configuration C_N and C_A are calculated as a function of α and M by Missile DATCOM™ for three times to obtain aerodynamic coefficient for fore X_{cg} location, aft X_{cg} location and after burnout. Then both C_N and C_A are linearly interpolated for the current M and α values.

2.1.3 Atmospheric Characteristics

Atmospheric density is a function of altitude as seen from equation 2.13.

$$\rho = \begin{cases} \rho_0 .(1 - 0.00002256.h)^{4.256} & \text{for } h \le 10000m. \\ 0.412.e^{-0.000151.(h-10000)} & \text{for } h > 10000m. \end{cases} \qquad (2.13)$$

where ρ is the atmospheric density which is calculated, h is the altitude and ρ_0 is the atmospheric density at sea level and it is equal to 1.223kg/m^3.

Aerodynamic coefficients depend also on Mach number. The Mach number is a dimensionless flow parameter and is defined as the ratio of the flow velocity to the speed of sound, c as,

$$M = \frac{V}{c} \qquad (2.14)$$

Where the speed of sound is calculated by;

$$c = \sqrt{(\gamma_{air} .R.T)} \qquad (2.15)$$

In this equation γ_{air}, is the specific heat ratio of the air, which is equal to 1.4, and R is the universal air gas constant, which is equal to 287 J/kg.K. T is the ambient temperature, which changes with altitude as;

$$T = \begin{cases} T_0 \cdot (1 - 0.00002256.h) & \text{for } h \le 10000m \\ 0.7744.T_0 & \text{for } h > 10000m \end{cases} \qquad (2.16)$$

Here, T_0 is the temperature at sea level and it is equal to 293 K.

2.1.4 Mass Characteristics

For an end-burning missile, mass and center of gravity data varies during flight due to propellant consumption. Total mass and X_{cg} values are obtained by the following formulas respectively:

$$m_{total} = \sum_{i=1}^{n} m_i \qquad (2.17)$$

$$X_{cg} = \frac{\sum_{i=1}^{n} m_i \cdot x_i}{m_{total}} \qquad (2.18)$$

where; m_i is the mass of the i[th] component, m_{total} total mass of the missile that includes the motor, warhead, inertial guidance unit, seeker, tails and wings and x_i is the location of the mass center of the i[th] component relative to a reference (in this case, the nose of the missile) [18].

The total mass of the missile is assumed to change linearly with time until burnout according to the following equation:

$$m = m_{total} - \frac{t}{t_b} m_{propellant} \qquad (2.19)$$

where, $m_{propellant}$ is the propellant mass, t the simulation time and t_b burnout time.

CHAPTER 3

GENETIC ALGORITHM TECHNIQUE FOR GLOBAL OPTIMIZATION

In this section Genetic algorithm is explained and then the computer program implemented algorithm is applied to the Zermelo's trajectory optimization problem. The improvements to the basic algorithm are proposed and tested.

3.1 Biological Background

The idea behind genetic algorithms is to do what nature does. They are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian strife for survival.

All living organisms consist of cells. In each cell there is the same set of chromosomes. Chromosomes are strings of DNA and serves as a model for the whole organism. A chromosome consists of genes, blocks of DNA. Each gene encodes a particular protein. Basically can be said, that each gene encodes a trait, for example color of eyes. Possible settings for a trait (e.g. blue, brown as for the color of eyes) are called alleles. Genes of certain characters are located at certain places of the chromosome , which are called loci (string positions).

Complete set of genetic material (all chromosomes) is called genome. Particular set of genes in genome is called genotype. The genotype is with later development after birth base for the organism's phenotype, its physical and mental characteristics, such as eye color, intelligence etc.

During reproduction, first occurs recombination (or crossover). Genes from parents form in some way the whole new chromosome. The new created offspring can then be mutated. Mutation means, that the elements of DNA are a bit changed. These changes are mainly caused by errors in copying genes from parents. The fitness of an organism is measured by success of the organism in its life [23].

3.2 Genetic Algorithm

A genetic algorithm for a particular problem must have the following components:

3.2.1 Representation

3.2.1.1 The binary representation

In the binary representation each element of a chromosome vector is coded using the same number of bits. To facilitate fast run-time decoding, each element occupies its own word of memory: this way elements can be accessed as integers, which removes the need for binary to decimal decoding. Then each chromosome is a vector of N words, which equals number of elements per chromosome. Binary genetic algorithm solves many optimization problems that stump traditional techniques. When the parameters are naturally quantized, the binary genetic algorithm fits nicely. However, when the parameters are continuous, it is much logical to represent them by floating point numbers. In addition, since the binary genetic algorithm has its precision limited by the binary representation of parameters, using real numbers instead easily allows representation to the machine precision [7].

3.2.1.2 The floating point representation

In the floating point (FP) representation each chromosome vector is coded as a vector of floating point numbers, of the same length as the solution vector. Each element is forced to be within the desired range, and the operators are carefully designed to preserve this requirement. The precision of such an approach depends on the underlying machine but is generally much better than of the binary representation Of course, we can always extend the precision of the binary representation by introducing more bits, but this considerably slows down the algorithm [17]. This floating point implementation also has the advantage of requiring less storage than the binary genetic algorithm because a single floating point number represents the parameter instead of N_{bits} integers. As N_{bits} increases, this storage becomes significant. Of course, the other advantage is in the accurate representation of the continuous parameter. It follows that the representation of the cost function is also more accurate as result [7].

In this thesis, floating point representation is preferred. Because the conducted experiments done by Michalewicz indicate that the floating point representation is faster, more consistent from run to run, and provides a higher precision. In addition, the floating point representation, as closer to the problem space, is easier for designing other operators incorporating problem specific knowledge.

3.2.2 Initial Population

Algorithm is started with a set of solutions (represented by chromosomes) called population. A population of chromosomes are initialized randomly. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness; the more suitable they are the more chances they have to reproduce.

3.2.3 Evaluation Function

Calculate the fitness value eval (v_i) for each chromosome v_i (i=1, ... , pop_size)

$$eval\ (v_i) = fitness\ of\ chromosome\ v_i \qquad (3.1)$$

3.2.4 Parameters

Population size (pop_size), probability of crossover and probability of mutation are the parameters to be determined.

3.2.4.1 Population Size

The population size dictates the number of chromosomes (or members) in the population (in one generation). If there are too few chromosomes, GA has a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to increase population size, because it does not help solving the problem faster. The best population size is both application dependent and related to the length of the chromosome. Many researchers suggest population sizes between 25 and 100.

3.2.4.2 Crossover Probability

Crossover probability says how often crossover will be performed. If there is no crossover, offspring is exact copy of parents. If there is a crossover, offspring is made from parts of parents' chromosome. If crossover probability is 100%, then all offspring is made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population [23].

Crossover is made in hope that new chromosomes will have good parts of old chromosomes and maybe the new chromosomes will be better. However it is good to leave some part of population survive to next generation.

3.2.4.3 Mutation Probability

Mutation probability says how often parts of chromosome will be mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed [23]. Large mutation rates increase the probability that good schemata will be destroyed, but increase population diversity. The best mutation rate is application dependent but for most applications is between 0.1% and 10%.

Mutation is made to prevent GA falling into local extreme, but it should not occur very often, because then GA will in fact change to random search.

3.2.5 Selection

Find the total fitness of the population [17]

$$F = \sum_{i=1}^{pop\_size} eval(v_i) \qquad (3.2)$$

Calculate the probability of a selection p_i for each chromosome v_i (i=1, ... , pop_size)

$$p_i = \text{eval (vi)} / \text{F} \qquad (3.3)$$

Calculate a cumulative probability q_i for each chromosome v_i (i=1, ... , pop_size)

$$q_i = \sum_{j=1}^{i} p_j \qquad (3.4)$$

The selection process is based on spinning the roulette wheel pop_size times; each time we select a single chromosome for a new population in the following way:

Generate a random (float) number r from the range [0..1].

If $r<q_1$ then select the first chromosome (v_1); otherwise select the i-th chromosome v_i ($2 \le i \le pop\_size$) such that $q_{i-1} < r \le q_i$.

3.2.6 Elitism

When creating new population by crossover and mutation, we have a big chance, that we will loose the best chromosome. Elitism is the name of method, which first copies the best chromosome (or a few best chromosomes) to the new population.

The rest is done in classical way. Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution [23].

3.2.7 Crossover

It is the linear combination of two vectors. It begins by randomly selecting a parameter in the first pair of the parents to be the crossover point [7].

$$\alpha = \text{roundup}\ \{\text{random}*\text{Npar}\} \qquad (3.5)$$

Let,

$$\text{Parent1}=[v_{m1} v_{m2} \ldots v_{m\alpha} \ldots v_{mNpar}] \qquad (3.6)$$

$$\text{Parent2}=[v_{d1} v_{d2} \ldots v_{d\alpha} \ldots v_{dNpar}] \qquad (3.7)$$

m and d subscripts discriminate between the mom and the dad parent. Then the selected parameters are combined to form new parameters that will appear in the children:

$$v_{new1}= v_{m\alpha}-\beta[\ v_{m\alpha}-\ v_{d\alpha}] \qquad (3.8)$$

$$v_{new2}= v_{d\alpha}+\beta[\ v_{m\alpha}-\ v_{d\alpha}]$$

where β is also a random value between 0 and 1. The final step is to complete the crossover with the rest of the chromosome.

$$\text{offspring1}=[\ v_{m1} v_{m2} \ldots v_{new1} \ldots v_{dNpar}] \qquad (3.9)$$

$$\text{offspring2}=[v_{d1} v_{d2} \ldots v_{new2} \ldots v_{mNpar}] \qquad (3.10)$$

If the first parameter of the chromosomes is selected, then only the parameters to the right of the selected parameter are swapped. If the last parameter of the chromosomes is selected, then only the parameters to the left of the selected parameter are swapped.

3.2.8 Mutation

If x_i^t=[$v_1,...,v_n$] is a chromosome, then each element v_k has exactly equal chance of undergoing the mutative process. The result of a single application of this operator is a vector [$v_1,...,v_k',...,v_n$] with $1 \le k \le n$, and v_k' a random value from the domain of the corresponding parameter domain$_k$ [17].

3.3 Sample Genetic Algorithm for Missile Trajectory Optimization

In this thesis, a genetic algorithm implemented in C programming language is used to solve missile trajectory optimization problems. In the following the structure of this algorithm is given in detail. The numerical values given in the explanations are the same as the ones used in the original code except for the fitness values. It is thought that the original values will be confusing and difficult to follow. So more simple fitness values are chosen for presentation while the idea remains the same.

The main structure of the code is taken from Michalewicz [17]. It is composed of 7 parts being initialize, evaluate, keep the best, select, crossover, mutate and elitist. Following is an explanation of the functions of each of those parts over a step-by-step track of the algorithm.

3.3.1 Initialize

The first thing necessary for a genetic evolution process to take place through generations is the existence of an initial population. Since in our case it is desired to have a missile trajectory path evolving to reach the largest distance in the minimum possible time, the members of our population must be different trajectory paths having the trajectory control parameters as the genes in their chromosome. The control parameters in our case are the angle of attack values used during the flight. 20 angle of attack values are used at equal time intervals up to the burnout of the engine fuel and 40 angle of attack values are used at equally spaced energy levels for the rest of the flight.

The angle of attack values for the first member of the initial population can be defined by the programmer so as to see the effects of different seeds. The rest of the initial population is formed in a random manner. The lower and the upper boundaries of the valid angle of attack values are stored in a text file (alpha.txt). First these boundaries are read from the file and the angle of attack values are generated for the remaining members of the initial population accordingly. Since it is not desired to have a greatly fluctuating, inconsistent angle of attack profile the variations between the consecutive angle of attack values are assigned as the genes rather than the actual angle of attack values themselves. Only the first angle of attack value (the first gene of the chromosome) is chosen among the full range of valid region and the subsequent angle of attack values are obtained adding the variations represented by the corresponding genes consecutively.

Similar to the original scenario the population consists of 49 members each having a single chromosome made up of 60 genes. Note that member(i) can be represented by chromosome(i) for i=1,2,...,49. Denoting gene(j) of chromosome(i) with $gene_{i,j}$ for i=1,2,...,49 and for j=1,2,...,60 the genetic map of the population can be represented in a tabular form as in Figure 3.1. The reciprocity of the flight mechanics terms in genetic algorithm terminology is tabulated in Table 3-1.

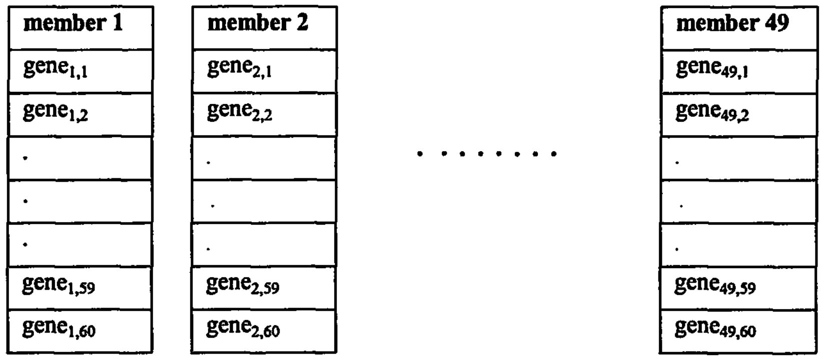


Figure 3.1 Initial Population of the Genetic Algorithm

Table 3-1 The Relation between the Terminologies of GA and Flight Mechanics

GA Terminology	**Flight Mechanics Terminology**
$gene_{i,j}$ (i= 1, 2, ..., 49) & (j= 1, 2, ..., 60) i represents the member in a population j represents the gene in the chromosome of a member	$\alpha_{i,j}$ (i= 1, 2, ..., 49) & (j= 1, 2, ..., 60) i represents the trajectory among the 49 sample trajectories j represents the AOA parameters in a trajectory
$chromosome_i$	$(\alpha_1, \alpha_2, ..., \alpha_{60})_i$
$member_i$ (same as chromosome since there is only one chromosome in each member)	$(\alpha_1, \alpha_2, ..., \alpha_{60})_i$ (same as the item above since a full set of AOA values is enough to represent a trajectory)

3.3.2 Evaluate

In this part of the code, the angle of attack values for each member of the population and the equations of motion of the missile are used to determine the flight trajectories for all the members and the corresponding fitness values are calculated. The fitness function is chosen such that the closer the desired parameters are approached the greater the results of the fitness function will get.

First the motor related parameters of the missile model are assigned and the initial setting of the variables are made according to the launch conditions. Before burnout, the height, velocity, flight path angle and range of the missile are calculated over 20 nodes equally spaced in time using the angle of attack values determined for each node previously. After burnout, the calculations are made over 40 nodes this time spaced at equal energy levels. In this process, total energy that is the sum of the kinetic and the potential energy of the missile is calculated just after the burnout and at the terminal point. Then until the missile energy reaches the energy at the terminal point the height, velocity, flight path angle and range of the missile are calculated again. The angle of attack input between nodes is obtained by linear interpolation.

The final results are used to obtain the fitness value of the flight trajectory. In Figure 3.2 the members and the corresponding fitness values are illustrated as follows;

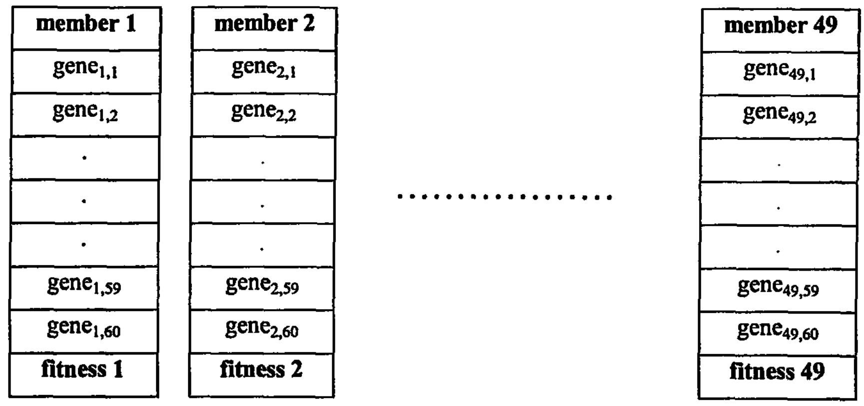


Figure 3.2 In Evaluate Part, Fitness Values of the Members are Calculated

3.3.3 Keep the Best Function

This function keeps track of the best member of the population. An additional 50^{th} entry in the population array holds a copy of the best individual.

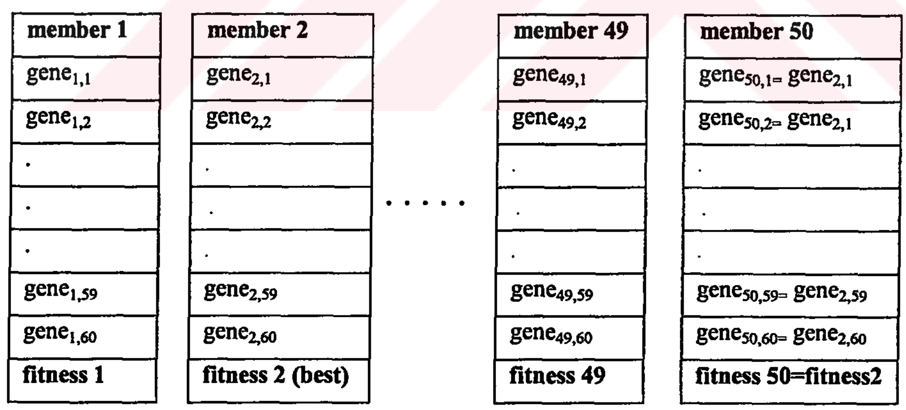


Figure 3.3 Last Member is the Best Member of the Population

3.3.4 Select

In this part of the code, cumulative fitness of the members in the population are calculated. Survivors of the population are selected using these cumulative fitness values to form up the parents for the second generation of the population. In chapter 3.2.5, the selection process is explained in detail. In Figure 3.4 below the fitness value of each member is given numerically to be used in the selection process.

member 1	member 2	member 3		member 48	member 49	member 50
gene$_{1,1}$	gene$_{2,1}$	gene$_{3,1}$		gene$_{48,1}$	gene$_{49,1}$	gene$_{50,1}$
gene$_{1,2}$	gene$_{2,2}$	gene$_{3,2}$		gene$_{48,2}$	gene$_{49,2}$	gene$_{50,2}$
.						
.	.	.	....	.	.	.
.						
gene$_{1,59}$	gene$_{2,59}$	gene$_{3,59}$		gene$_{48,59}$	gene$_{49,59}$	gene$_{50,59}$
gene$_{1,60}$	gene$_{2,60}$	gene$_{3,60}$		gene$_{48,60}$	gene$_{49,60}$	gene$_{50,60}$
9	**100**	**18**		**3**	**32**	**100**

Figure 3.4 Members of the First Generation (Generation$_1$)

In this population, the best member is member$_2$ and worst member is member$_{48}$. "member$_{50}$" is the copy of the best member, member$_2$.

Now, the parents for the next generation (generation$_2$) will be selected among the members of the previous generation according to their cumulative fitness values as follows. "member$_{50}$" does not enter this selection process; it is kept without being changed to be used in the elitist process of the next generation.

1. Find the total fitness of the previous generation:

total fitness = 9 + 100 + 18 +....+ 3 + 32 =2800

2. Calculate relative fitness values:

$Rfitness_1$ = 9 / 2800 = 0.00321

$Rfitness_2$ = 100 / 2800 = 0.03571

$Rfitness_3$ = 18 / 2800 = 0.00643

...

$Rfitness_{48}$ = 3 / 2800 = 0.00107

$Rfitness_{49}$ = 32 / 2800 = 0.01143

3. Calculate cumulative fitness values:

$Cfitness_1 = Rfitness_1 = 0.00321$

$Cfitness_2 = Cfitness_1 + Rfitness_2 = 0.03892$

$Cfitness_3 = Cfitness_2 + Rfitness_3 = 0.04535$

......

$Cfitness_{48} = Cfitness_{47} + Rfitness_{48} = 0.98857$

$Cfitness_{49} = Cfitness_{48} + Rfitness_{49} = 1.00000$

4. Select survivors using cumulative fitness:

p is a random number between 0 and 1

let p_1= 0.002

$p_1 \leq \text{Cfitness}_1$ So;

$\text{Newfitness}_1 = 9$ $\text{Newmember}_1 = \{gene_{1,1}, gene_{1,2},..., gene_{1,60}\}$

let p_2= 0.04

$Cfitness_2 \leq p_2 \leq Cfitness_3$ So;

$\text{Newfitness}_2 = 18$ $\text{Newmember}_2 = \{gene_{3,1}, gene_{3,2},..., gene_{3,60}\}$

Let p_3=0.02

$Cfitness_1 \leq p_3 \leq Cfitness_2$ So;

$\text{Newfitness}_3 = 100$ $\text{Newmember}_3 = \{gene_{2,1}, gene_{2,2},..., gene_{2,60}\}$

...

Let p_{49}=0.58

$Cfitness_{26} \leq p_{49} \leq Cfitness_{27}$ So;

$\text{Newfitness}_{49} = 21$ $\text{Newmember}_{49} = \{gene_{27,1}, gene_{27,2},..., gene_{27,60}\}$

The parent members of the next (second) generation is obtained as follows;

member 1	member 2	member 3		member 48	member 49
$gene_{1,1}$	$gene_{2,1}$	$gene_{3,1}$		$gene_{48,1}$	$gene_{49,1}$
$gene_{1,2}$	$gene_{2,2}$	$gene_{3,2}$		$gene_{48,2}$	$gene_{49,2}$
.	.	.		.	.
.	.	.	. . . .	.	.
.	.	.		.	.
$gene_{1,59}$	$gene_{2,59}$	$gene_{3,59}$		$gene_{48,59}$	$gene_{49,59}$
$gene_{1,60}$	$gene_{2,60}$	$gene_{3,60}$		$gene_{48,60}$	$gene_{49,60}$
9	18	100		100	21

Figure 3.5 Selected Parents for the Second Generation

3.3.5 Crossover

In this process first, parent members which will take part in crossover are selected, then the chromosomes of the selected pairs of parents are crossed over while the other parents are just passed over to the next process without any change. Heuristic crossover, which is explained in Chapter 3.2.7, is used in this optimization problem.

First of all, crossover probability should be determined. This can be constant for all generations or can be varied during the optimization process. Crossover probabilities used in various runs of the algorithm are shown in Table 3-2.

Table 3-2 Constant and Variable Crossover Probabilities Used during the Optimization

	Number of generations	Crossover prob
Constant crossover probability	$\max.gen.num$	80%
Variable Crossover Probability	$gen < 20\% of \max.gen.num$	80%
	$20\% of \max.gen.num < gen \leq 40\% of \max.gen.num$	75%
	$40\% of \max.gen.num < gen \leq 60\% of \max.gen.num$	70%
	$60\% of \max.gen.num < gen \leq 80\% of \max.gen.num$	65%
	$gen > 80\% of \max.gen.num$	50%

Let crossover probability be %75 (0.75).

A random number, x is chosen between 0 and 1 for each parent member.

let $x_1 = 0.62$ for the first run.

$x_1 \leq 0.75$ so, $member_1$ is selected as the first parent to crossover

let $x_2 = 0.85$ for the second run.

$x_2 \geq 0.75$ so, $member_2$ is not selected.

let $x_3 = 0.62$ for the third run.

$x_3 \leq 0.75$ so, $member_3$ is selected as the second parent to crossover

So, $member_1$ and $member_3$ are selected as the first pair of parents for crossover.

$member_1 = \{gene_{1,1}, gene_{1,2},..., gene_{1,60}\}$

$member_3 = \{gene_{3,1}, gene_{3,2},..., gene_{3,60}\}$

According to Heuristic crossover let β =32, which is the synopsis point that chromosomes of the two selected members will be swapped. So new offspring become as follows;

$Newmember_1 = \{gene_{1,1}, gene_{1,2},...,\mathbf{newgene_{1,32}}, gene_{3,33},..........., gene_{3,59}, gene_{3,60}\}$

$Newmember_3 = \{gene_{3,1}, gene_{3,2},...,\mathbf{newgene_{3,32}}, gene_{1,33},..........., gene_{1,59}, gene_{1,60}\}$

Then, other members are selected and new offspring are formed with the same process.

3.3.6 Mutation

Random uniform mutation is used for this optimization problem. A gene of a chromosome selected for mutation is replaced by a random value between lower and upper bounds of this gene.

Same as the crossover, a mutation probability number should be chosen first. As in the case of crossover this number can be a constant or can be varied during the optimization process.

Table 3-3 Constant and Variable Mutation Probabilities Used During the Optimization

	Number of generations	Mutation prob
Constant mutation probability	$\max.gen.num$	10%
Variable mutation Probability	$gen < 20\% of \max.gen.num$	10%
	$20\% of \max.gen.num < gen \leq 40\% of \max.gen.num$	4%
	$40\% of \max.gen.num < gen \leq 60\% of \max.gen.num$	2%
	$60\% of \max.gen.num < gen \leq 80\% of \max.gen.num$	1.5%
	$gen > 80\% of \max.gen.num$	1%

Let mutation probability be 0.08

x is again a random number between 0 and 1 decided for each gene of each member separately.

let $x_{1,1} = 0.2$ for the first run.

$x_{1,1} \geq 0.08$ so $gene_1$ in $chromosome_1$ **is not selected** for mutation

.

let $x_{3,30}=0.05$ for the 150th run (30th gene in 3rd chromosome, 60+60+30=150)

$x_{3,30} \leq 0.08$ so $gene_{3,30}$ **is selected** for mutation.

This process goes on until all the members' genes are checked for mutation..

After all the crossover and the mutation processes are applied, new population becomes ready as follows.

member 1	member 2	member 3		member 48	member 49	member 50
$gene_{1,1}$	$gene_{2,1}$	$gene_{3,1}$		$gene_{48,1}$	$gene_{49,1}$	$gene_{50,1}$
$gene_{1,2}$	$gene_{2,2}$	$gene_{3,2}$		$gene_{48,2}$	$gene_{49,2}$	$gene_{50,2}$
.	.	.		.	.	.
.	.	*$newgene_{3,30}$*	...	.	.	.
.	.	.		.	.	.
$Gene_{3,59}$	$gene_{2,59}$	$gene_{1,59}$		$gene_{48,59}$	$gene_{28,59}$	$gene_{50,59}$
$Gene_{3.60}$	$gene_{2,60}$	$gene_{1,60}$		$gene_{48,60}$	$gene_{28,60}$	$gene_{50,60}$

Figure 3.6 New Population After the Crossover and Mutation Processes are Applied

This population is used as the candidate parents for the third generation after the evaluation process is applied and new fitness values are calculated

member 1	member 2	member 3		member 48	member 49	member 50
gene$_{1,1}$	gene$_{2,1}$	gene$_{3,1}$		gene$_{48,1}$	gene$_{49,1}$	gene$_{50,1}$
gene$_{1,2}$	gene$_{2,2}$	gene$_{3,2}$		gene$_{48,2}$	gene$_{49,2}$	gene$_{50,2}$
.	.	.		.	.	.
.	.	.	. . .	.	.	.
.	.	.		.	.	.
gene$_{1,59}$	gene$_{2,59}$	gene$_{3,59}$		gene$_{48,59}$	gene$_{49,59}$	gene$_{50,59}$
gene$_{1,60}$	gene$_{2,60}$	gene$_{3,60}$		gene$_{48,60}$	gene$_{49,60}$	gene$_{50,60}$
48	**18**	**63**		**100**	**105**	**100**

Figure 3.7 Population of the New Generation

3.3.7 Elitism

The best member of the previous generation is stored as the last in the array. If the best member of the current generation were worse, then the best member of the previous generation would replace the worst member of the current population. As the best member of the current generation is better than the last member in this case, the last member in the array is replaced by the best member of the current generation.

member 1	member 2	member 3		member 48	member 49	member 50
gene$_{1,1}$	gene$_{2,1}$	gene$_{3,1}$		gene$_{48,1}$	gene$_{49,1}$	gene$_{49,1}$
gene$_{1,2}$	gene$_{2,2}$	gene$_{3,2}$		gene$_{48,2}$	gene$_{49,2}$	gene$_{49,2}$
.	.	.		.	.	.
.	.	.	. . .	.	.	.
.	.	.		.	.	.
gene$_{1,59}$	gene$_{2,59}$	gene$_{3,59}$		gene$_{48,59}$	gene$_{49,59}$	gene$_{49,59}$
gene$_{1,60}$	gene$_{2,60}$	gene$_{3,60}$		gene$_{48,60}$	gene$_{49,60}$	gene$_{49,60}$
48	**18**	**63**		**100**	**105**	**105**

Figure 3.8 Elitism is Applied to the Population

It is seen that $member_{49}$ has a fitness greater than the $member_{50}$' fitness. So we replace $member_{50}$' chromosome with the $member_{49}$' chromosome.

By applying crossover and mutation, new generations are formed until the optimum one is reached.

3.4 Zermelo's Trajectory Optimization Problem

In this problem as described by Bryson [21], a ship must travel through a river where the current varies with (x, y) location. Given any initial point (x_0, y_0), the trajectory of a ship is to be found that will place it at the final point (x_f, y_f) in minimum time. It is assumed that the ship travels at a constant speed V.

$$u = u(x, y)\ ,\ v = v(x, y) \tag{3.11}$$

(u, v) are the velocity components of the current in the x and y directions, respectively.

Equations of motion for the ship in the coordinate system are;

$$\dot{x} = V\cos\theta + u(x, y) \tag{3.12}$$

$$\dot{y} = V\sin\theta + v(x, y) \tag{3.13}$$

where θ is the heading angle of the ship's axis relative to the coordinate axis as shown in Figure 3.9.

Let's simplify this problem such that the current flows in the x direction and that the speed of the current varies linearly with y position.

$$u = -V\,(y/h),\ v = 0 \tag{3.14}$$

If we choose V=1, h=1 for simplicity, then;

$$\dot{x} = \cos\theta - y \tag{3.15}$$

$$\dot{y} = \sin\theta \tag{3.16}$$

For a given initial point (x_0, y_0) = (3.66, -1.86) as in [21], we would like to find the trajectory of the ship that will place it at the origin (0. 0).

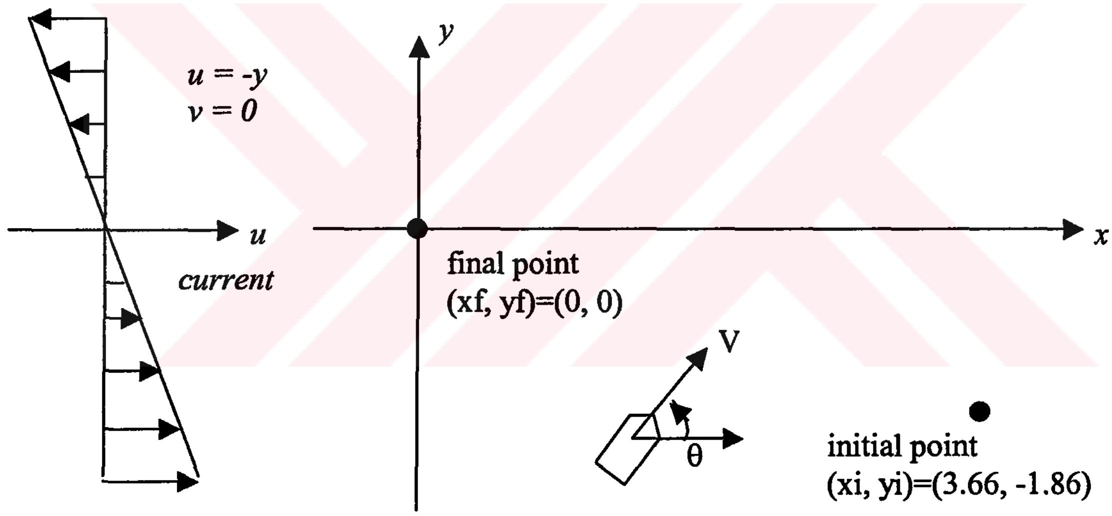


Figure 3.9 Zermelo's Trajectory Optimization Problem

Analytical solution which is obtained by Bryson for this problem is as follows:

$$\theta_i = 105^\circ, \theta_f = 240^\circ \qquad (3.17)$$

The time to go from the initial point to the origin is:

$t_f - t_i =$ 5.46 sec as tabulated in Table 3-4.

Table 3-4 Final Results of Analytical Solution

t_f	5.46 sec
x_f	0.000000
y_f	0.000000
θ_i	105°
θ_f	240°

3.5 Genetic Algorithm Results for Zermelo's Problem

In this part Zermelo's optimal control problem is converted to nonlinear programming problem by using Genetic algorithm as follows;

$$minimize\ J(p)$$

$$subject\ to\ G(p) = 0$$

$$p_l \leq p \leq p_u$$

where J is the cost function and equals to the time to reach origin t_f, p is the optimization parameters and includes the steering angles θ_i (i=1,2, ..., N) and t_f, G is the equality constraint and equals to range error [13].

N is the number of evenly-spaced time nodal points on the trajectory. The optimization parameters are total time and the value of the steering angles at these nodes.

At other time points, the value of steering angles are found by linear interpolation.

Penalty function approach is the most common approach in the genetic algorithms community [17]. So, it is also used in this optimization problem.

$$f = -t_f - k\left(\sqrt{x(t_f)^2 + y(t_f)^2}\right) \tag{3.18}$$

where k>0 is the penalty coefficient and f is the function that is optimized to find global maximum, $x(t_f)$ and $y(t_f)$ are the position of the ship at final time and t_f is the time to reach the origin.

The set of steering angles (θ) are specified at 10 evenly-spaced time points. Between these nodes steering angles are linearly interpolated. There are 30 members in one population. After 10,000th generation is calculated, the optimization ends. Initial starting point (3.66, -1.86) is same as in the analytical solution. In the first member of the first generation, initial steering angle at the nodes are taken as 0° and time to reach the origin is taken as 5 sec. These parameters are tabulated in Table 3-5. According to these parameters, initial trajectory is obtained as in Figure 3.10.

Table 3-5 Initial Parameters Used in Genetic Algorithm

Number of interval, N	10
Initial θ_i i=(1,2,...,10)	0°
Initial t_f	5 sec
x_i	3.66
y_i	-1.86

At the end of the algorithm final results are obtained in Table 3-6. According to these parameters final trajectory is obtained as in Figure 3.11.

Table 3-6 Results of Genetic Algorithm

t_f	5.461 sec
x_f	0.002116
y_f	0.003652
θ_i	101.368°
θ_f	241.02°

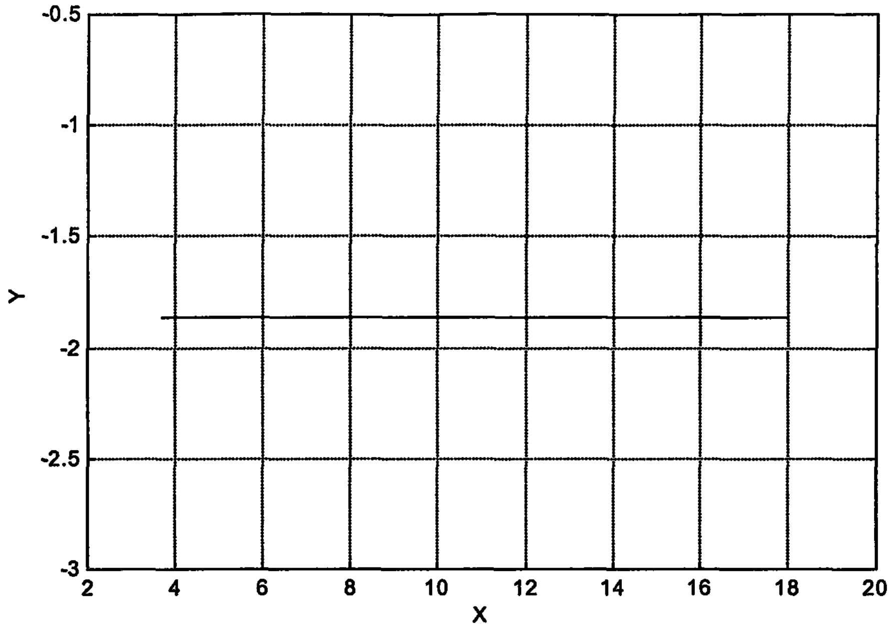


Figure 3.10 Initial Trajectory of the Zermelo Optimization Problem

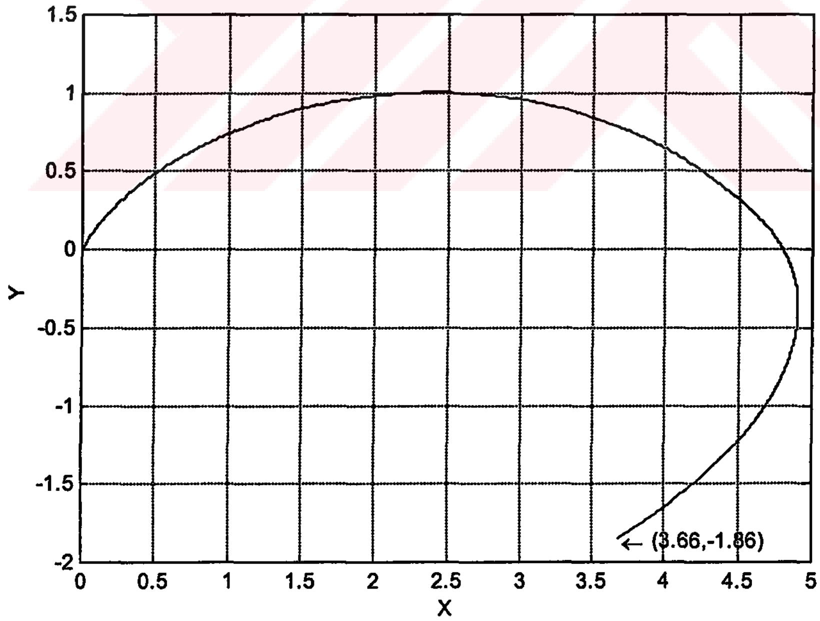


Figure 3.11 Final Trajectory of the Zermelo Optimization Problem

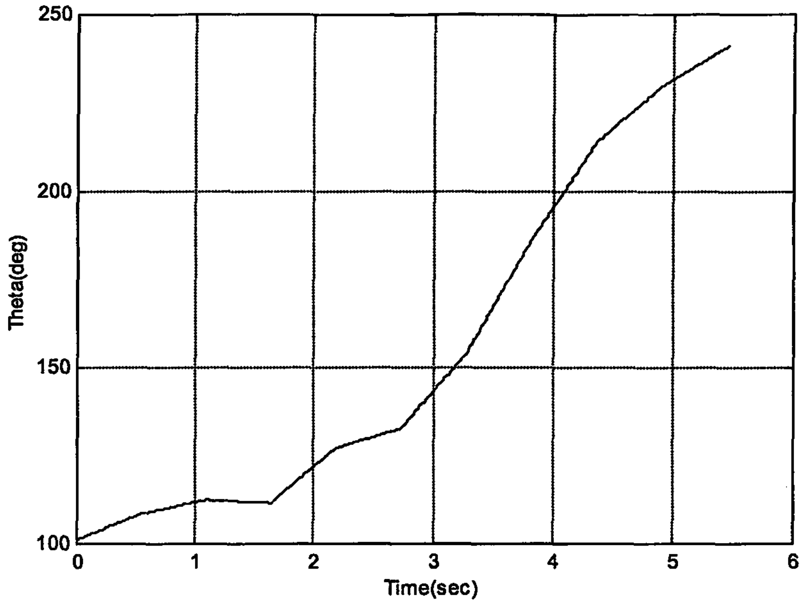
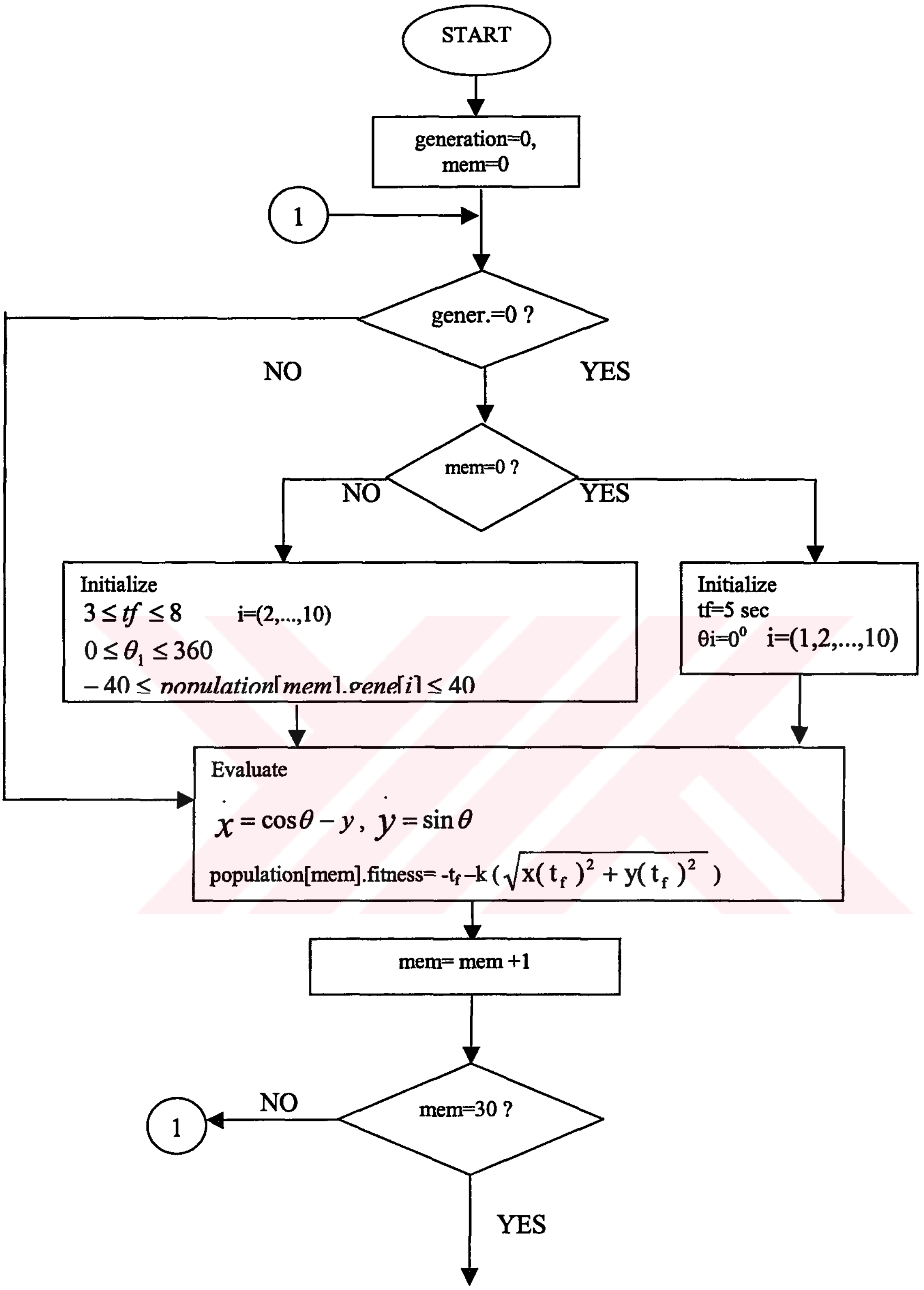


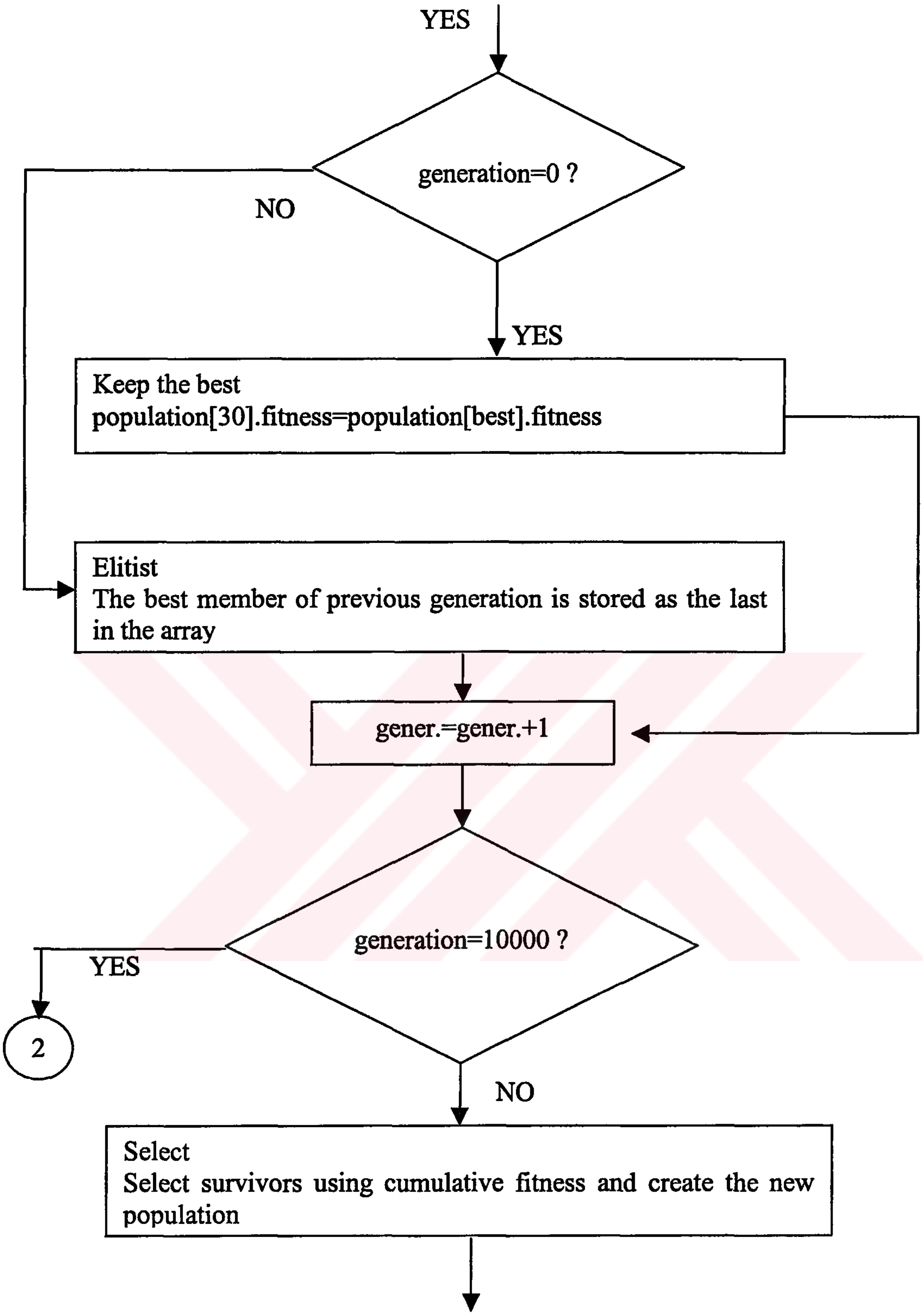
Figure 3.12 Final Steering Angle-Time Graph of Zermelo Optimization Problem

Final values obtained by the Genetic algorithm is similar to the analytical solution. According to these results, it is concluded that, genetic algorithm is efficient to be used for finding the solutions of trajectory optimization problems From Figure 3.13, flowchart of the algoritm used in this problem can be examined .



Flowchart of Genetic Algorithm on Zermelo's Optimal Control Problem

(Continued on Next Page)



Flowchart of Genetic Algorithm on Zermelo's Optimal Control Problem

(Continued on Next Page)

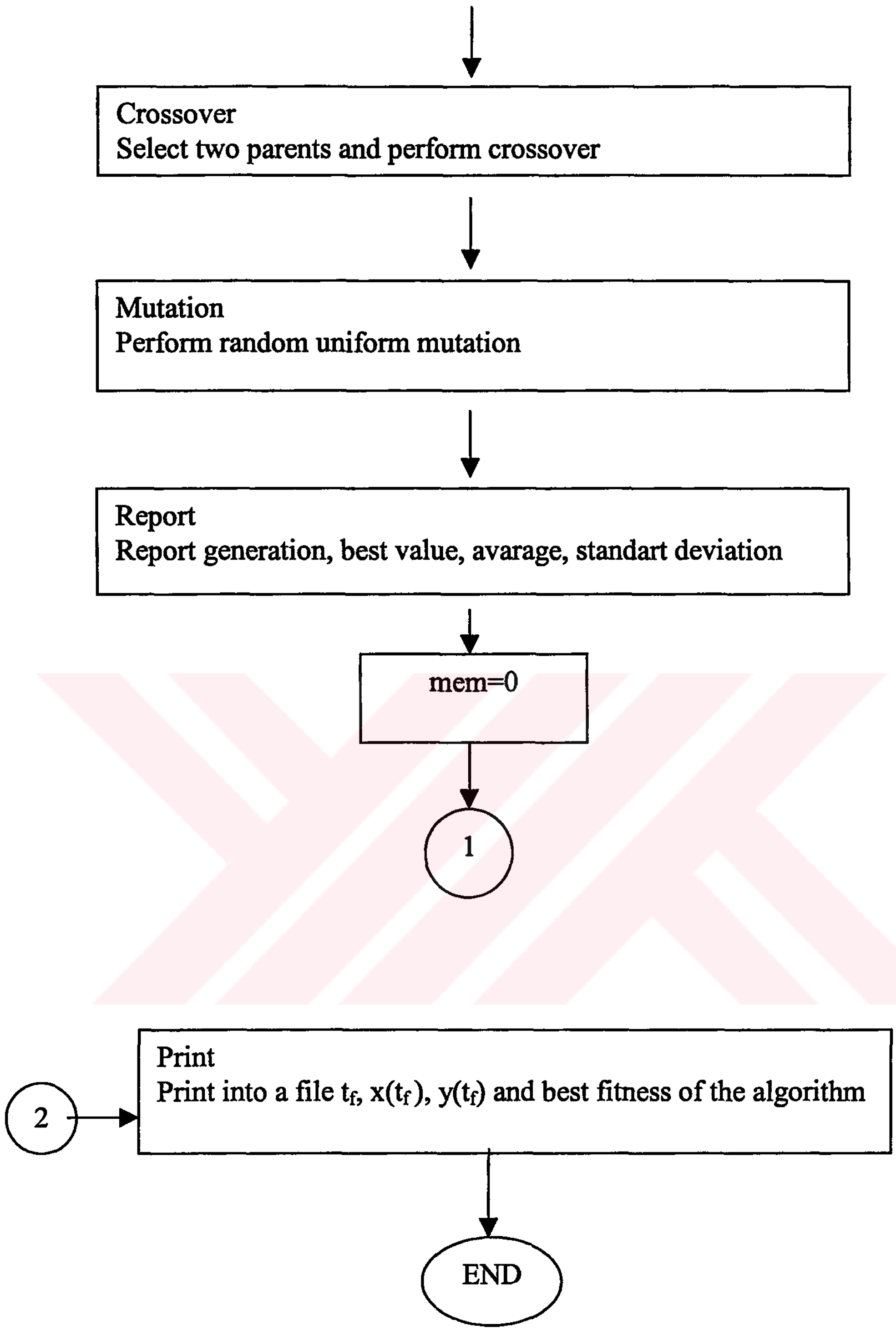


Figure 3.13 Flowchart of Genetic Algorithm on Zermelo's Optimal Control Problem

CHAPTER 4

AIR-TO-SURFACE MISSILE DESIGN OPTIMIZATION PROBLEMS

The design of missiles is an interesting and comprehensive field which requires a reasonably broad knowledge of the fundamentals of many technical specialties such as aerodynamics, kinematics, propulsion, structural design etc. The optimization of design is gained by careful analysis of the following considerations; Simplicity in external configuration to reduce development time and cost. Efficient aerodynamic control surfaces to simplify control and guidance system circuits and to minimize servo power requirements. Missile range, speed and other performance characteristics that satisfy the mission requirements. Simple, efficient, and highly reliable power plant. Adequacy of the airframe from the standpoint of stability, maneuverability, and dynamic responses.

In this study, the aim is to find the optimum flight path to carry a given missile to the maximum range, for a given launch and specified impact conditions.

In this chapter, the results of the parametric study which is expected to present some insight to the characteristics of the problem and to the method employed is presented.

In this vein, first maximum range problem is presented. Then some results on, specified range minimum flight time are given. The results of two formulations, namely, time interpolation and energy interpolation are compared. Finally a parametric study on optimization parameters such as population size, number of genes in each chromosome, cross-over probability are investigated.

This study is on finding the best trajectory of a missile guided in the pitch plane. The angle of attack is the input parameter. Since it is not possible to realize the desired angle of attack exactly due to measurement noise as well as the neglected pitch dynamics of the missile, it is decided to examine what happens when the angle of attack parameter is disturbed slightly from its desired value. The simulation conducted to examine the effect of imperfect angle of attack realization is presented in the final section.

4.1 Maximum Range Problem

The trajectory optimization problem for this study is defined as finding the missile flight path which gives the maximum range for the given launch and the impact conditions. Objective of the optimization is to find the optimal control histories for an air-to-surface missile to maximize the flight range for a specified target. Thus, the cost function of minimization is;

$$J = -Range \qquad (4.1)$$

subject to some equality constraints on the impact conditions;

$$g_1(\alpha(t_f)) = 0 \qquad g_i(x(t_f)) = 0 \quad i = 2,3,4 \qquad (4.2)$$

where, $x(t)$ is the state of the system, $\alpha(t)$ is the angle of attack in time and t_f is the final flight time. The procedure is based on assigning a set of angle of attack values for a number of sub-intervals of the total flight and then evaluating the success of the angle of attack profile depending on the results obtained finally. The nodes that define the sub-intervals are equally spaced in time up to burnout and at equal energy consumption intervals after burnout. The angle of attack values between the nodes are obtained by linear interpolation using the values at the nodes.

The dynamic equations used involve four state variables for two-dimensional motion in vertical plane, representing position, velocity, flight path angle and the angle of attack as the control parameter. These equations are presented in Chapter 2.

Through numerical integration of the dynamic equations of the missile, the original optimal control problem now becomes a nonlinear programming problem. The missile shall reach the desired target with specified impact conditions. Thus, the nonlinear programming problem is:

Maximize Range,

Subject to;

$$\gamma(t_f) = \gamma_f \qquad (4.3)$$

$$h(t_f) = h_f \qquad (4.4)$$

$$\nu(t_f) = \nu_f \qquad (4.5)$$

It is difficult beside being usually meaningless to exactly satisfy the above equality constraints. It is always reasonable to replace the equality constraints with tight inequality constraints and gives better results [13].

$$\gamma_f - a < \gamma(t_f) < \gamma_f + a \quad (4.6)$$

$$h_f - b < h(t_f) < h_f + b \quad (4.7)$$

$$\nu_f - c < \nu(t_f) < \nu_f + c \quad (4.8)$$

These tight inequality constraints may be combined with the objective function using Lagrange multipliers [13].

$$\begin{aligned} f = Range &+ k_1.\max(((\gamma(t_f) - \gamma_f)^2 - a^2),0) \\ &+ k_2.\max(((v(t_f) - v_f)^2 - b^2),0) + k_3.(h(t_f) - h_f)^2 \end{aligned} \quad (4.9)$$

where, $k_1 = k_2 = -2000$ and $k_3 = -20$ are penalty coefficients, h_f=1000m, γ_f=-75° and ν_f=271.2 m/sec are the desired terminal conditions on the altitude, the flight path angle and the velocity respectively, a=±0.2° , b= ±0.2 m/s and c=0 are the tight inequality constraints.

The launch conditions of a hypothetical air-to-surface missile are given in Table 4-1.

Table 4-1 Launch Conditions of Air-to-Surface Missile

Velocity (V_0)	0.7 mach = 237.3 m/sec
Flight path angle (γ_0)	5°
Altitude (h_0)	8000 m

In the motor model, total missile mass m_t and propellant mass m_p are calculated for thrust and burnout time values tabulated in Table 4-2.

Table 4-2 ASM Motor Related Parameters

Thrust, T	6000 N
Burnout time, t_b	70 sec
Total missile weight, m_t	598.975 kg
Propellant mass, m_p	166.576 kg

Angle of attack history is parameterized by 20 nodes at boost and 40 nodes at coast. To start the optimization an initial angle of attack history is used. The initial angle of attack set is chosen to be 1° at all nodes. The simulation results using this initial angle of attack set are given in figures 4.1 to 4.5. The total flight time of the missile is found to be 51.2 sec., which is even less than the powered stage time of the missile. As the angle of attack values are not high enough for the missile to gain altitude, missile starts to fall down. Finally, the range of the missile at 1000 m altitude is obtained as 17402 m, the impact angle is $-31^{\circ\circ}$ and the Mach number is 1.1453.

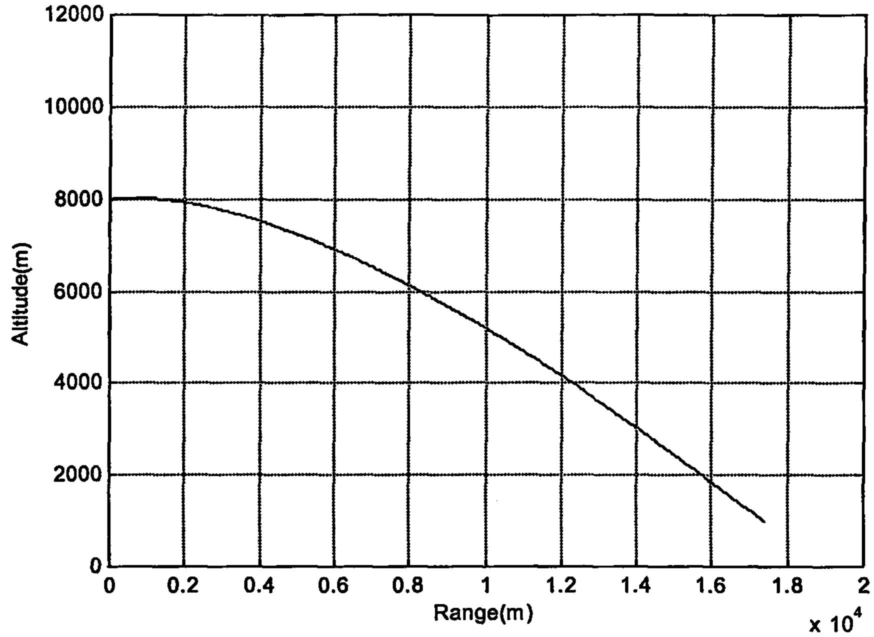


Figure 4.1 Initial, Altitude versus Range Graph

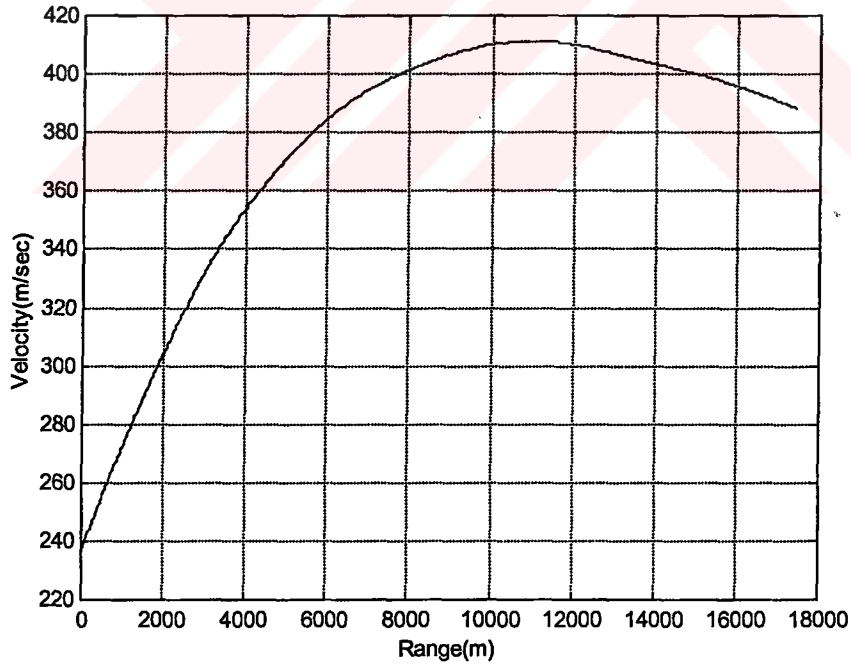


Figure 4.2 Initial, Velocity versus Range Graph

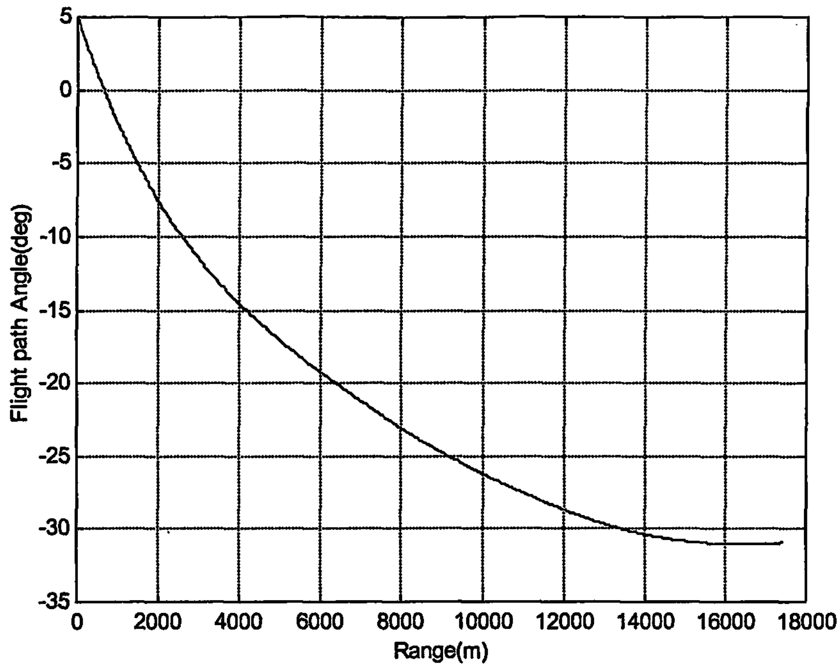


Figure 4.3 Initial, Flight Path Angle versus Range Graph

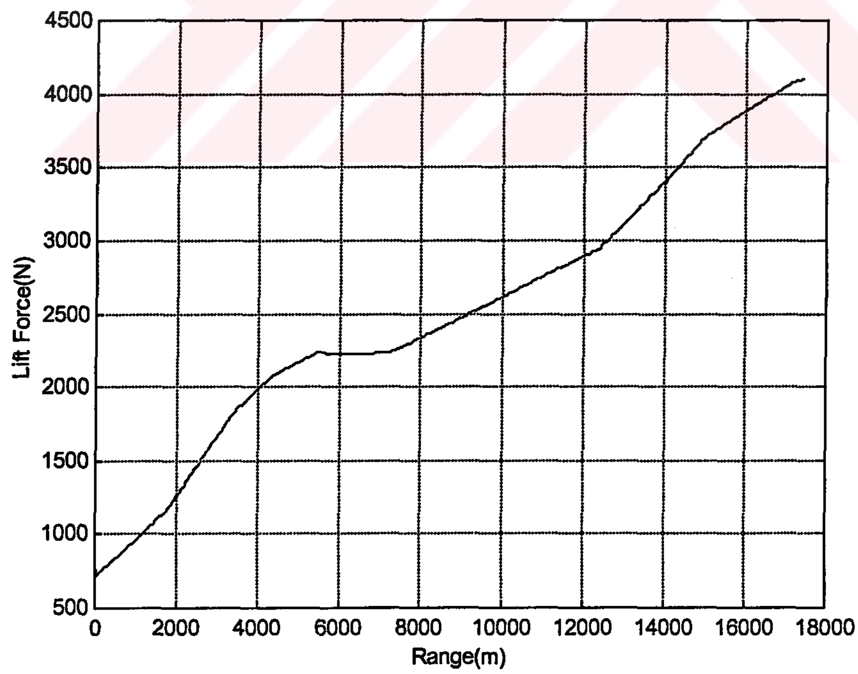


Figure 4.4 Initial, Lift Force versus Range Graph

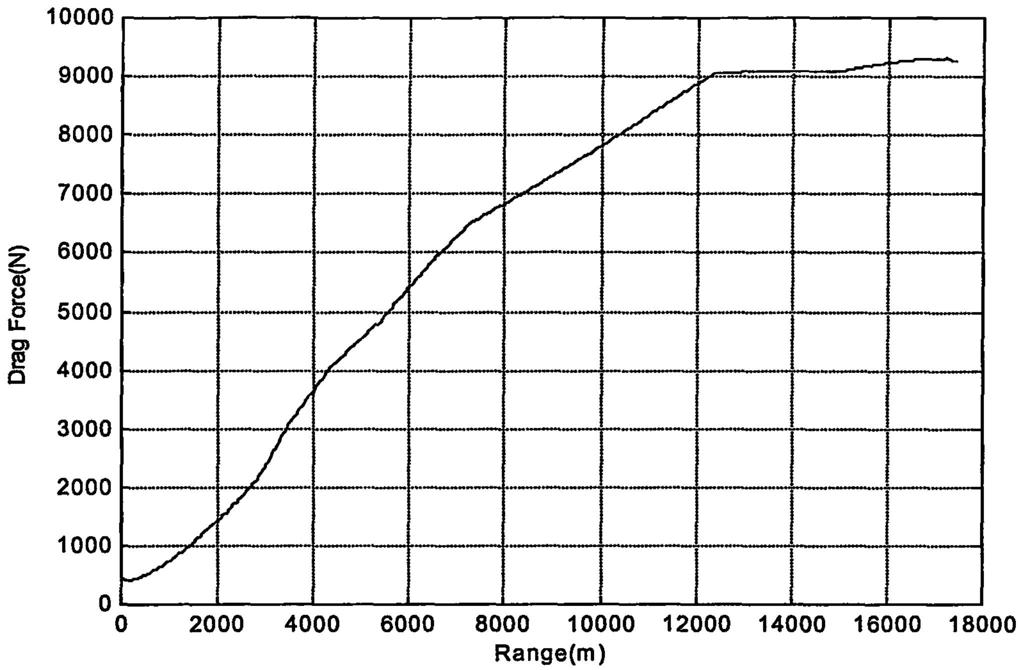


Figure 4.5 Initial, Drag Force versus Range Graph

For operational success, the missile impact conditions are very important [4]. For this purpose the terminal constraints given in Table 4-3 are specified.

Table 4-3 Terminal Conditions for ASM

Velocity (V_f)	0.8 mach = 271.2 m/sec
Flight path angle (γ_f)	-75°
Altitude (h_f)	1000 m
Angle of attack (α_f)	0°

The maximum range obtained from the maximum range trajectory optimization problem solution is 81,582m with very closely satisfied terminal conditions. Let's compare this with the missile's ballistic trajectory. The ballistic trajectory results are given in figures 4.6 to 4.10. In a ballistic trajectory the angle of attack throughout the trajectory is zero, and so the lift force. The range obtained by the ballistic flight is 13,247 m at an altitude of 27 m, with a final mach number of 1.16 and flight path angle of –52.68°. Total flight time is 43.2 sec, which is even less than burnout time.

To verify the simulation program; these ballistic trajectory results are compared with the ballistic trajectory results given in reference [13] and listed in Table 4-4. The small variations may be caused by quantization errors and the results are very close.

Table 4-4 Comparison of Ballistic Trajectory Results

	Ballistic trajectory obtained by the current program	Ballistic trajectory reported in [13]
Final range (m)	13247 m	13250 m
Final Height (m)	27 m	39 m
Final Mach number	1.16	1.16
Final flight path angle	-52.68°	-52.7 °
Total flight time (sec)	43.2 sec	43.16 sec

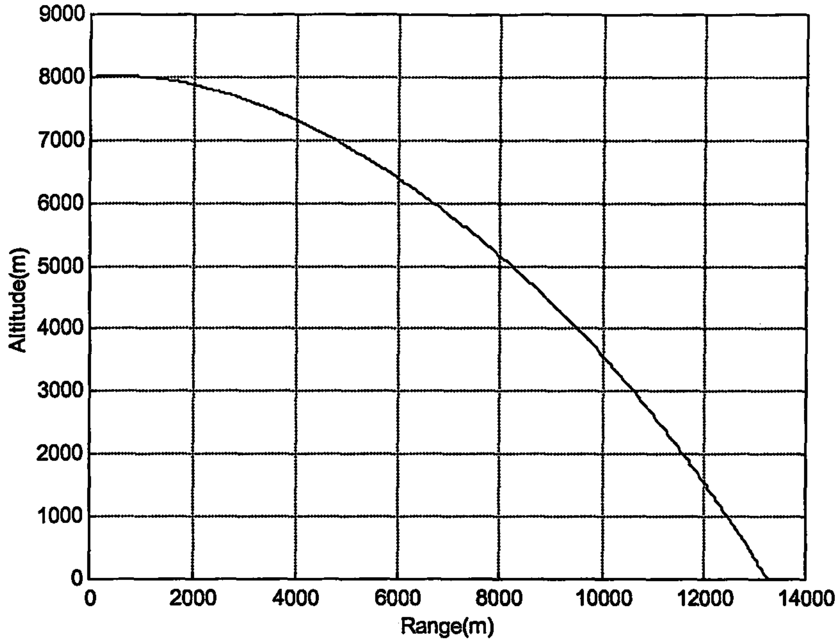


Figure 4.6 Ballistic Trajectory, Altitude versus Range Graph

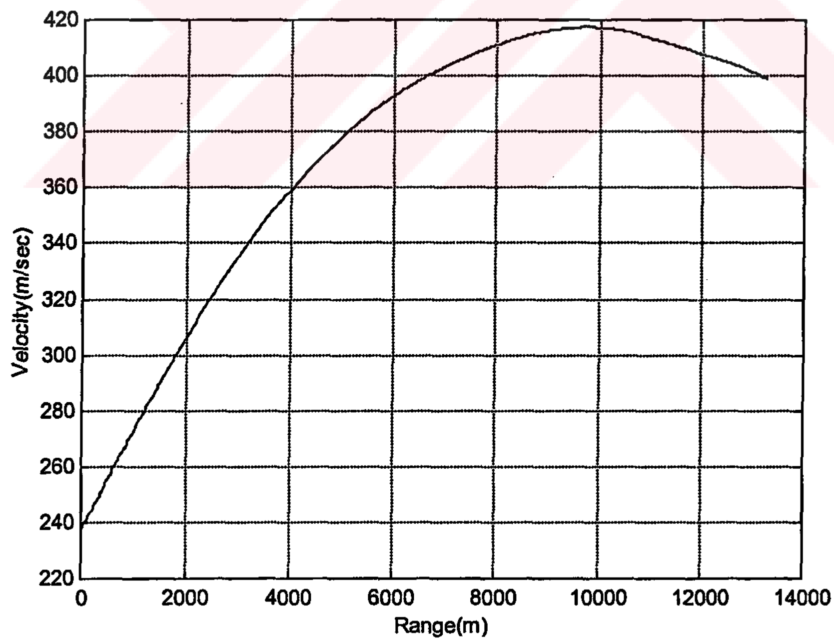


Figure 4.7 Ballistic Trajectory, Velocity versus Range Graph

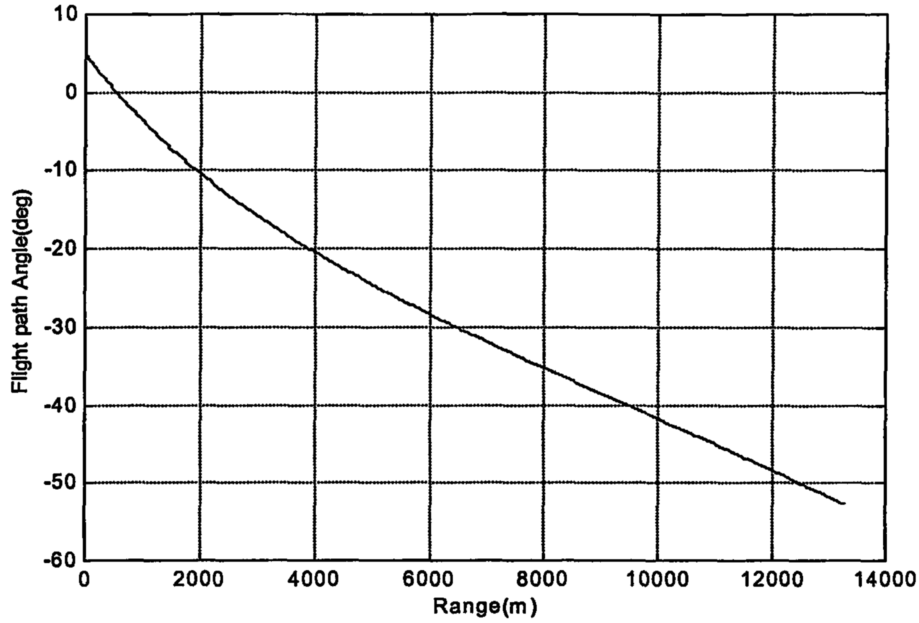


Figure 4.8 Ballistic Trajectory, Flight Path Angle versus Range Graph

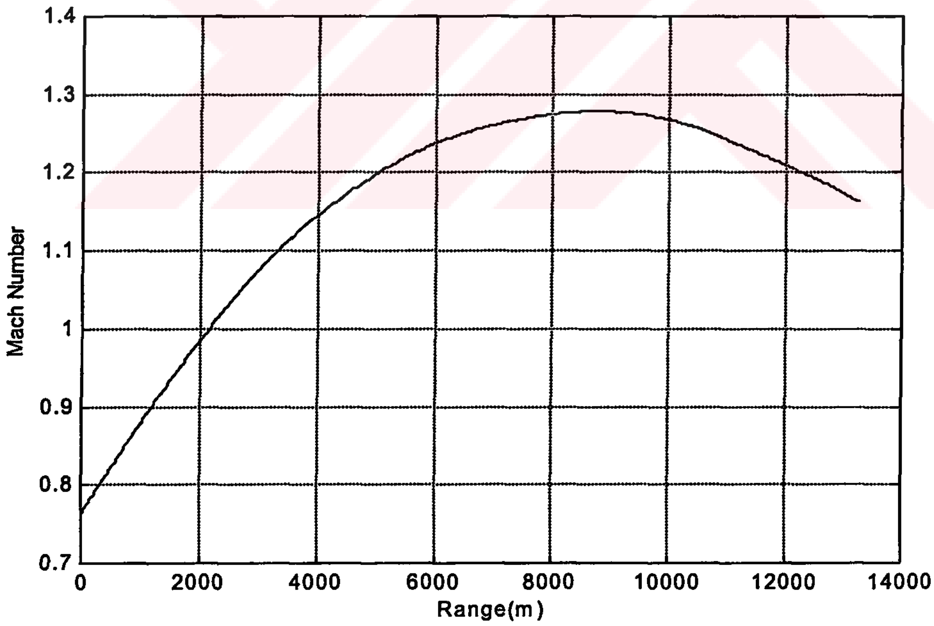


Figure 4.9 Ballistic Trajectory, Mach Number versus Range Graph

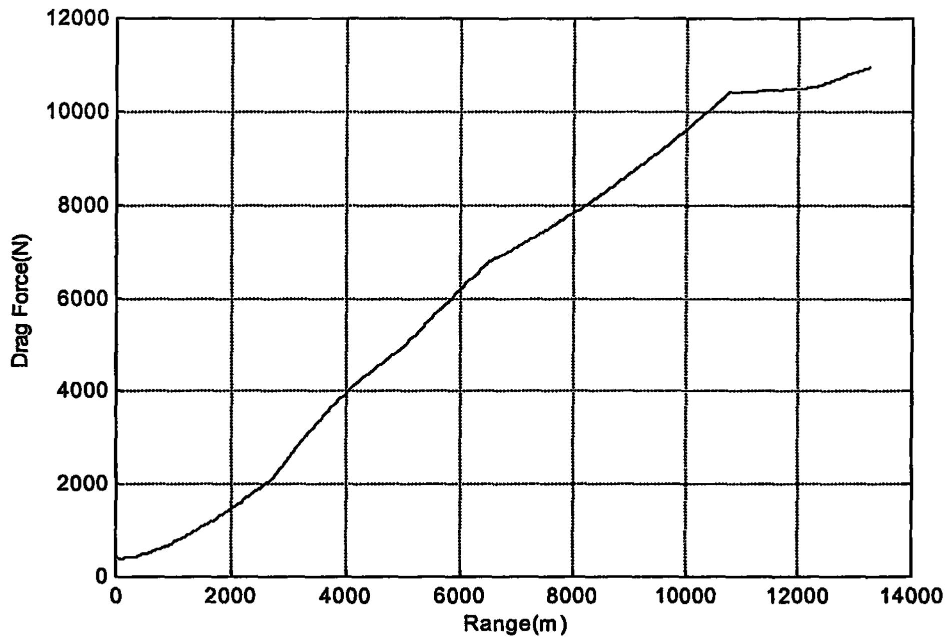


Figure 4.10 Ballistic Trajectory, Drag Force versus Range Graph

The maximum range trajectory obtained using the optimization code is given in Figure 4.13. The angle of attack history with respect to flight time and range are presented in Figure 4.11 and Figure 4.12. The graphs of velocity and flight path angle versus range are given in, Figure 4.14 and 4.15 respectively. Drag and Lift forces are plotted against range in Figure 4.16 and 4.17. Velocity changes are also shown in terms of Mach number in Figure 4.18.

In the optimization there were 50 members in each population and the evolution was continued for 1000 generations. The maximum range at 1000m altitude is obtained as 81582m. The final velocity is 271.1 m/sec and the flight path angle is –74.91°, which are within the boundaries of the desired terminal conditions. Angle of attack is parameterized by 20 nodes before and 40 nodes after burnout. The angle of attack values are bounded between -8° and 8° due to the limits on control actuation system. These values decrease up to burnout because lift is enough to reach high altitude. But after burnout, the angle of attack values increase to 8° and stays around this value then, a sharp decrease occurs to satisfy the impact condition. Missile needs high angle of attack values to increase the lift force which is needed to reach a higher range. Velocity (as well as Mach number) of the missile increases continuously until burnout. After burnout as the thrust drops to zero suddenly, there is a sharp decrease in the velocity (or Mach number) of the missile. When the altitude of the missile reaches its maximum value, flight path angle becomes zero and then starts to decrease since missile dives. But after 50 kilometers, there is an increase in the flight path angle. This can be again to increase its lift to reach the desired range.

In Table 4-5, simulated annealing and genetic algorithm results for maximum range optimization problem are listed. The simulated annealing results are plotted together with GA results in Appendix-B. In simulated annealing algorithm [13] 30 angle of attack values are used during the maximum range optimization problem. In genetic algorithm, 60 angle of attack values (60 genes) are used. Both algorithms reached the desired terminal conditions. But maximum range obtained by GA is higher than the range obtained by SA. It is decided to check what happens when 30 angle of attack values (30 genes) are used in the genetic algorithm. After 10,000 generations the maximum range obtained is only 75,654 m. This shows that 30 genes are not enough to optimize the trajectory to have a resulting range around 80 km.

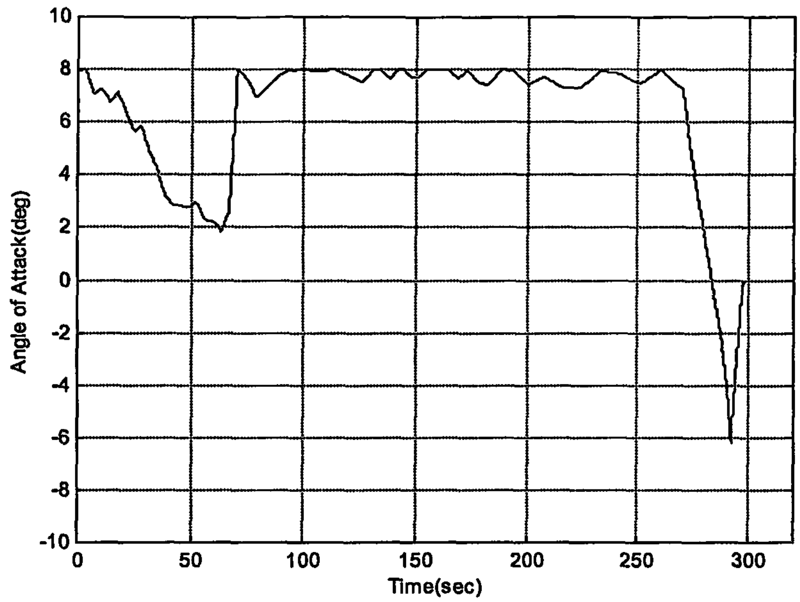


Figure 4.11 Maximum Range, Angle of Attack versus Flight Time Graph

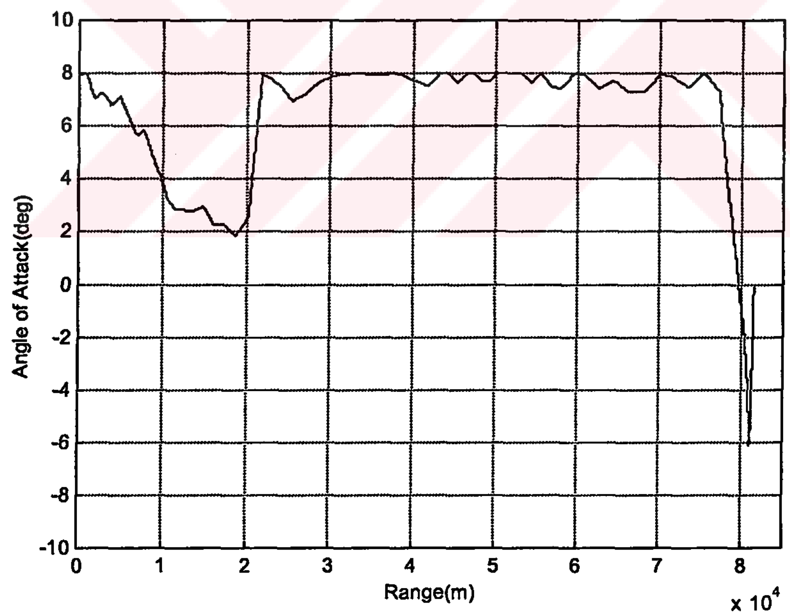


Figure 4.12 Maximum Range, Angle of Attack versus Range Graph

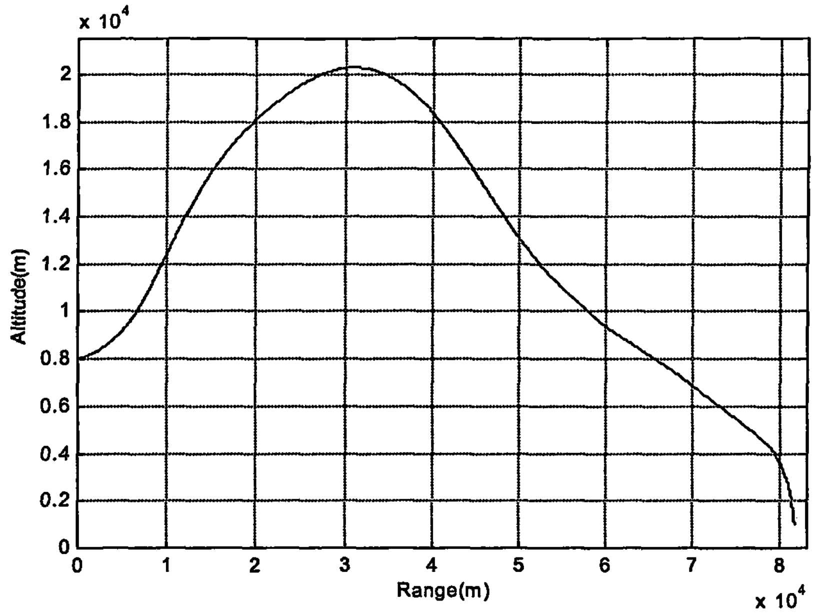


Figure 4.13 Maximum Range, Altitude versus Range Graph

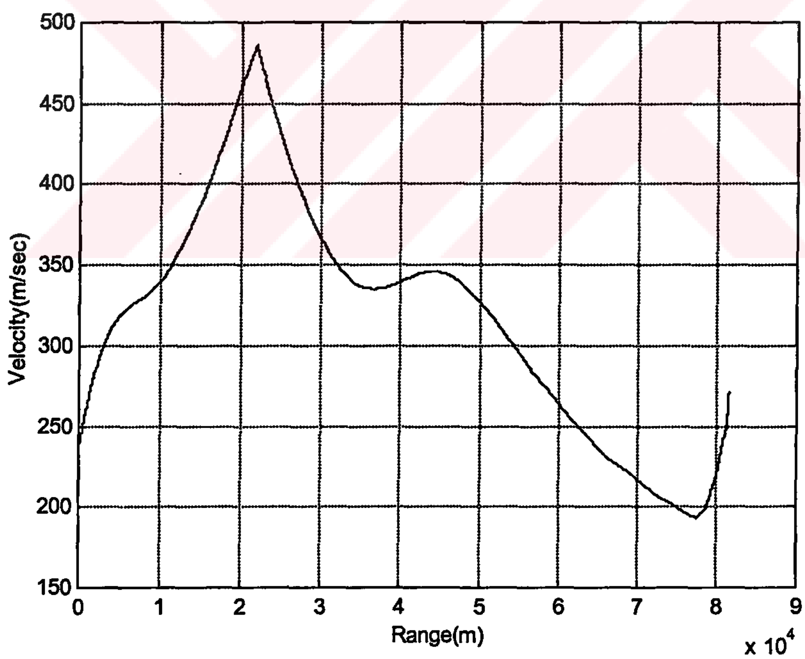


Figure 4.14 Maximum Range, Velocity versus Range Graph

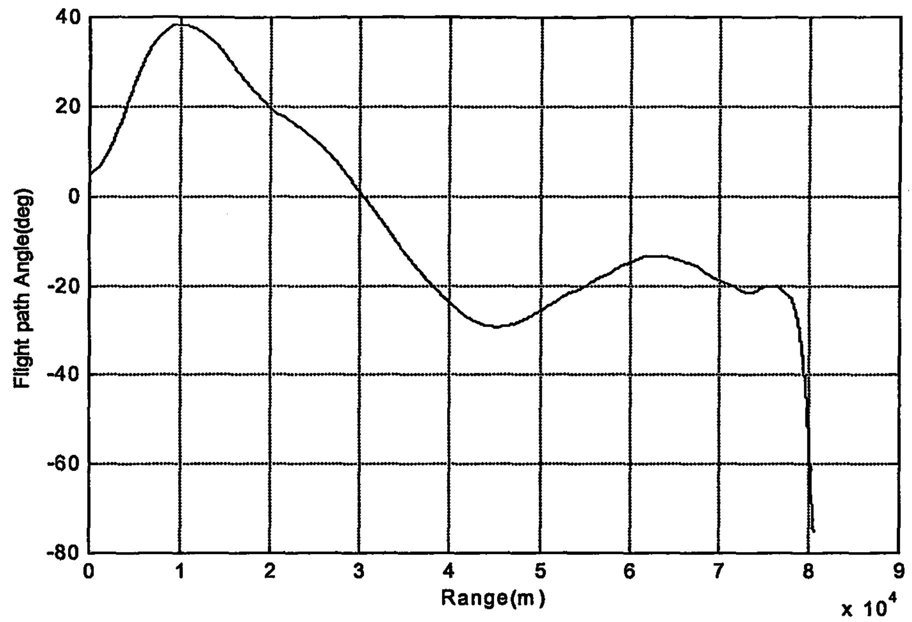


Figure 4.15 Maximum Range, Flight Path Angle versus Range Graph

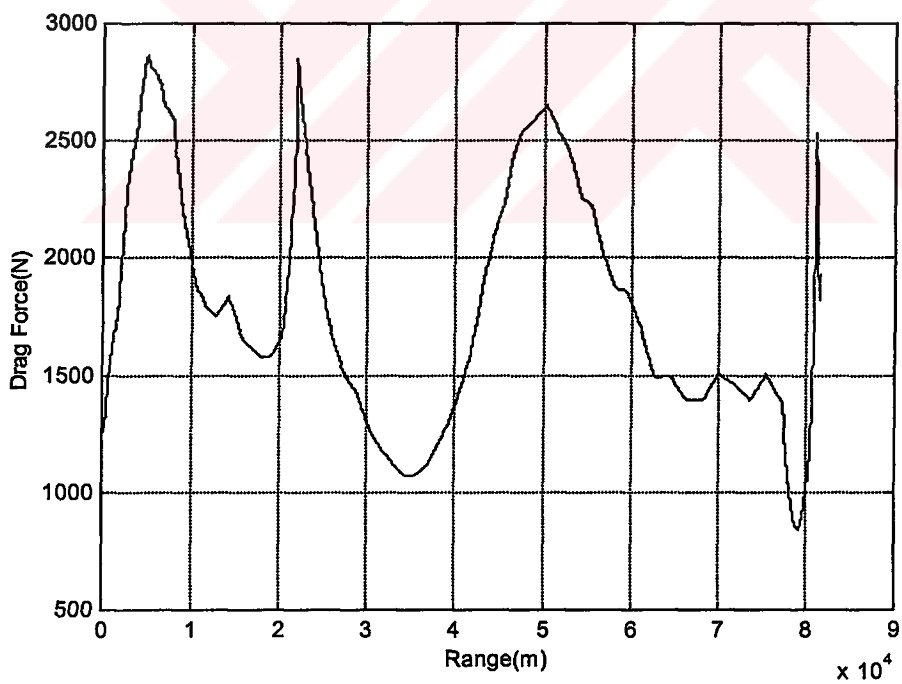


Figure 4.16 Maximum Range, Drag Force versus Range Graph

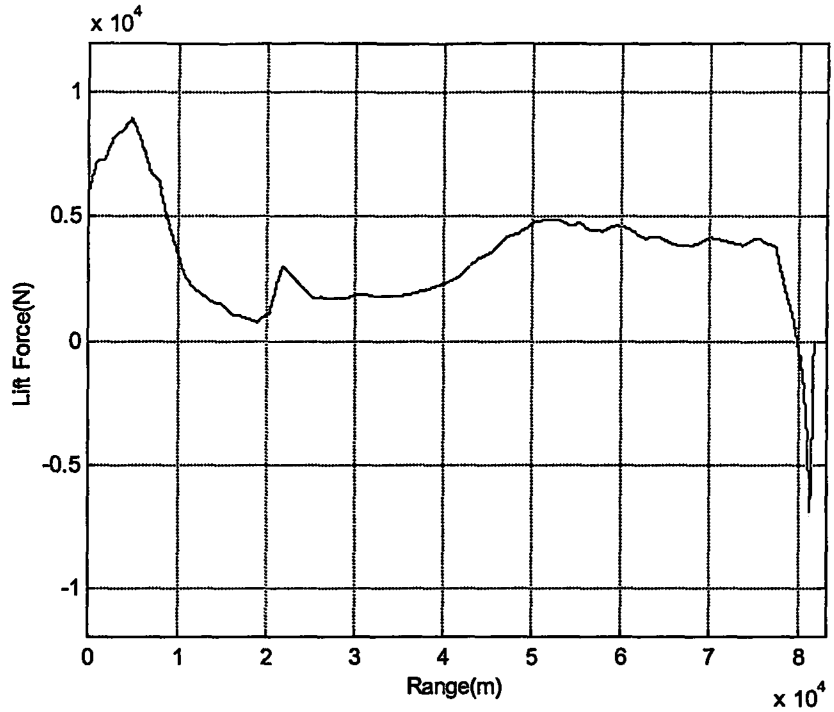


Figure 4.17 Maximum Range, Lift Force versus Range Graph

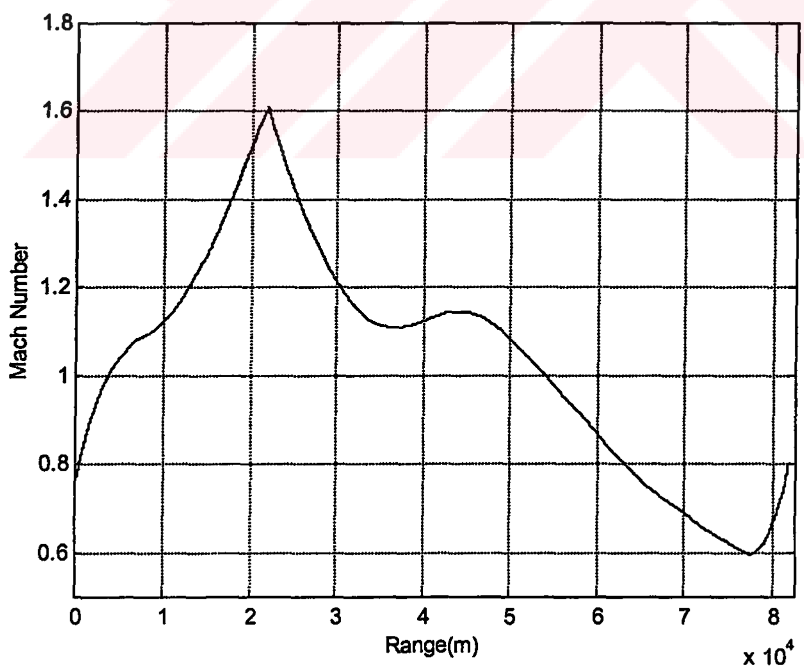


Figure 4.18 Maximum Range, Mach Number versus Range Graph

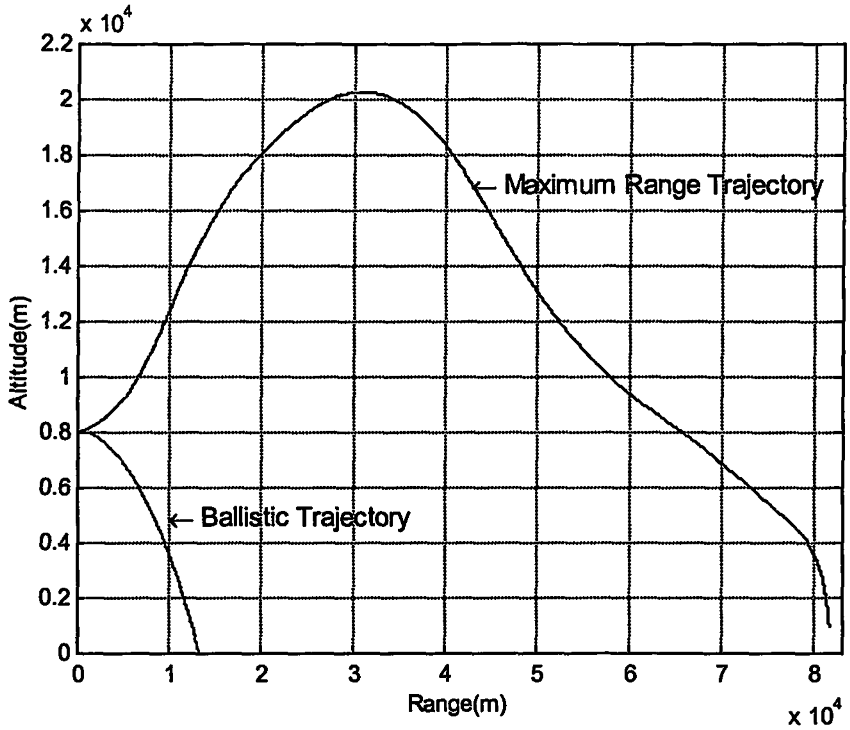


Figure 4.19 Ballistic & Maximum Range Trajectory Comparison Graph

Table 4-5 Final Results of Maximum Range Problem for GA & SA

	GA Results (30 genes)	**GA Results (60 genes)**	**SA results (30 A.O.A)**
Final range, $r(t_f)$	75654 m	81582 m	73250 m
Final altitude, $h(t_f)$	1000m	999.5m	1000m
Final mach number, $M(t_f)$	0.80	0.80	0.80
Final flight path angle, $\gamma(t_f)$	-75.16°	-74.91°	-74.99°
Final total flight time, t_f	269.2 sec	297.5 sec	257.4 sec

The angle of attack values obtained from the maximum range trajectory optimization for 30 genes are used as input parameters for the SA algorithm. The results are tabulated in Table 4-6. The trajectory obtained is plotted together with the original GA output in Figure 4.20.

Table 4-6 Simulation Results of SA Code Using the Optimal A.O.A Values Obtained by GA

	GA Results (30 genes)	SA results (30 A.O.A)
Final range, $r(t_f)$	75654 m	75596 m
Final altitude, $h(t_f)$	1000m	1059 m
Final mach number, $M(t_f)$	0.80	0.79
Final flight path angle, $\gamma(t_f)$	-75.16°	-76.89°

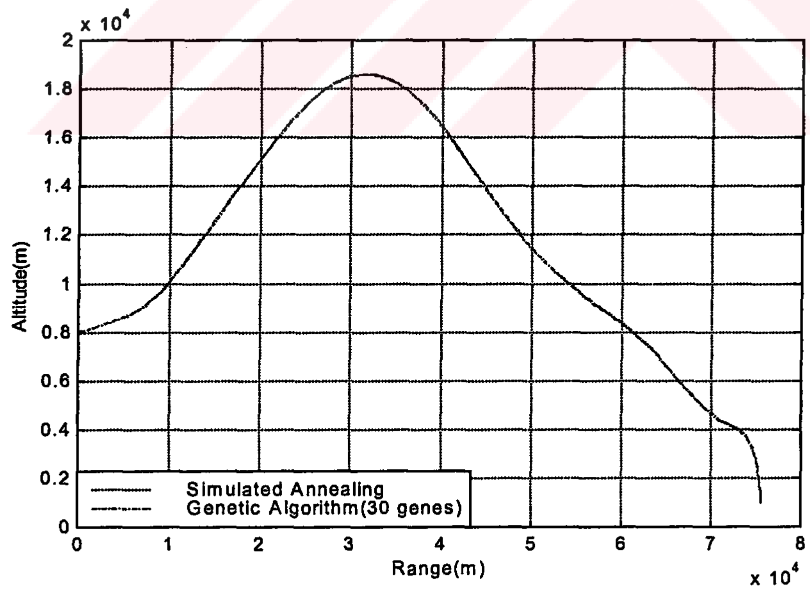


Figure 4.20 Comparison of the Missile Flight Mechanics Simulation Program of Bingöl to the Simulation Program Developed for this Study

4.2 Minimum Time Problem

Objective of the minimum time optimization is to find the optimal control histories for an air-to-surface missile to hit a target at a specified range in minimum time with given launch and specified impact conditions. Thus, the function to be minimized is

$$J = Time \qquad (4.10)$$

subject to some equality constraints on the trajectory

$$g_1(\alpha(t_f)) = 0 \qquad g_i(x(t_f)) = 0 \quad i = 2,3,4,5 \qquad (4.11)$$

where, $x(t)$ is the state of the system, $\alpha(t)$ is the angle of attack in time and t_f is the final flight time. Again the optimization parameters are the angle of attack inputs at the nodes as before.

Again the penalty function is used to handle the equality constraints, such that;

$$\begin{aligned} f &= k_1.Time + \mathrm{k}_2.\max(((\gamma(t_f) - \gamma_f)^2 - a^2),0) \\ &\quad + k_3.\max(((v(t_f) - v_f)^2 - b^2),0) + k_4.(\,h(t_f) - h_f)^2 - r(t_f - r_f)^2 \end{aligned} \qquad (4.12)$$

where, k_1=-10000, k_2 =k_3 =-2000 and k_3=-20 are penalty coefficients, h_f=1000m, γ_f=-75° , v_f=271.2 m/s and rf= 73000m are the desired terminal conditions on the altitude, the flight path angle, velocity and the range respectively, a=±2° , b= ±2 m/s and c=0 are the tight inequality constraints.

The launch conditions and the motor model of the air-to-surface missile are same as the one used in maximum range optimization problem of this missile. Terminal conditions are tabulated in Table 4-7.

The desired range is specified to be 73000m which is lower than the maximum range. Angle of attack history is parameterized by 20 nodes for the duration of boost and 40 nodes for the duration of coast. These nodes are equally spaced in time before burnout and equally spaced in energy after burnout. Between the nodes angle of attack inputs are linearly interpolated. To start the optimization an initial angle of attack history is used which is the same as the one used in maximum range optimization problem.

Table 4-7 Terminal Conditions for ASM

Velocity (V_f)	0.8 mach = 271.2 m/sec
Flight path angle (γ_f)	-75°
Altitude (h_f)	1000 m
Angle of attack (α_f)	0°
Final Range (r_f)	73000 m

Minimum flight time optimization problem results are given in figures 4.23 to 4.29. The angle of attack history with respect to flight time is presented in Figure 4.21. The graphs of altitude, velocity and flight path angle versus range are given in Figure 4.22, Figure 4.23 and Figure 4.24 respectively. Drag and Lift forces are plotted against range in Figure 4.25 and Figure 4.26. Velocity changes are also shown in terms of Mach number in Figure 4.27. Minimum flight time optimization results are plotted together with the maximum range optimization results in Appendix-B.

There are 50 members in each population with 60 genes and the evolution is continued for 5000 generations. In Table 4-8, it is seen that, terminal conditions become more accurate and flight time decreases when the generation number increases,. The minimum time obtained from the optimization problem solution is 242.2 sec with the satisfied impact conditions.

Table 4-8 Change of Flight Time with the Increasing Number of Generation

Generation number	100	200	500	1000	5000
Range (m)	72941	73000	72966	72952	72982
Flight path angle (deg)	-66.04	-68.9	-72.99	-73.08	-73.05
Velocity (m/sec)	269.89	271.1	271.4	271.08	271.20
Flight time (sec)	**244.3**	**246.5**	**245.6**	**245**	**242.2**

The results found at the end of the optimization are listed in Table 4-9. The angle of attack values are bounded between -8° and 8° due to the limits on control actuation system. These values decrease up to burnout because lift is enough to reach high altitude. But after burnout, missile increases the angle of attack values to reach the decided range. But the angle of attack values are not as high as the values in the maximum range problem because the specified range is 73 kilometers which is lower than the maximum range. Also for the same reason, the maximum altitude that the missile reaches in this problem is lower than the altitude in the maximum range problem.

Velocity (as well as Mach number) of the missile increases continuously until burnout. After burnout as the thrust drops to zero suddenly, there is a sharp decrease in the velocity (or Mach number) of the missile. The increase in the velocity after 70 kilometers is to satisfy the impact conditions. When the altitude of the missile reaches its maximum value, flight path angle becomes zero and then starts to decrease.

Table 4-9 Final Results of Minimum Time Problem

Final range, $r(t_f)$	72982 m
Final altitude, $h(t_f)$	999 m
Final velocity, $V(t_f)$	271.2 m/sec
Final flight path angle, $\gamma(t_f)$	-73.0°
Final angle of attack, $\alpha(t_f)$	-0.15°
Final total flight time, t_f	242.2 sec

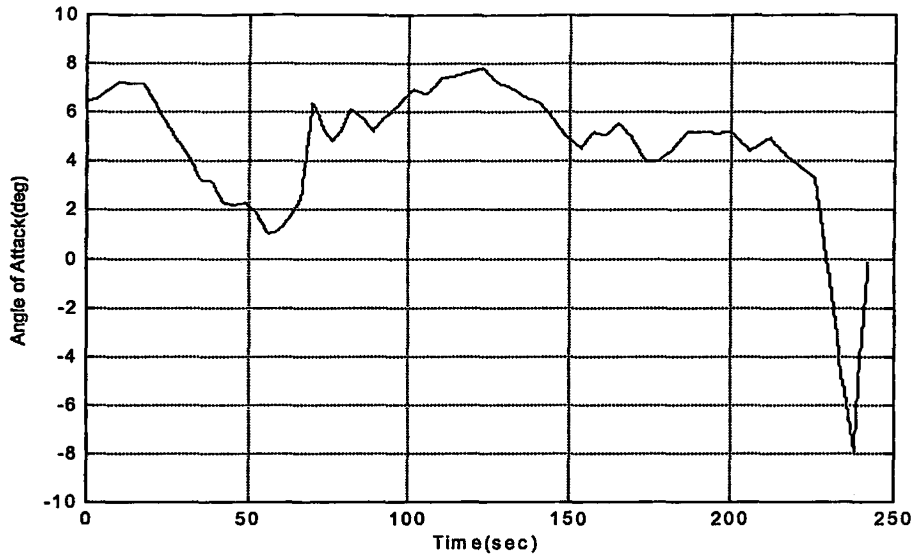


Figure 4.21 Minimum Time, Angle of Attack versus Flight Time Graph

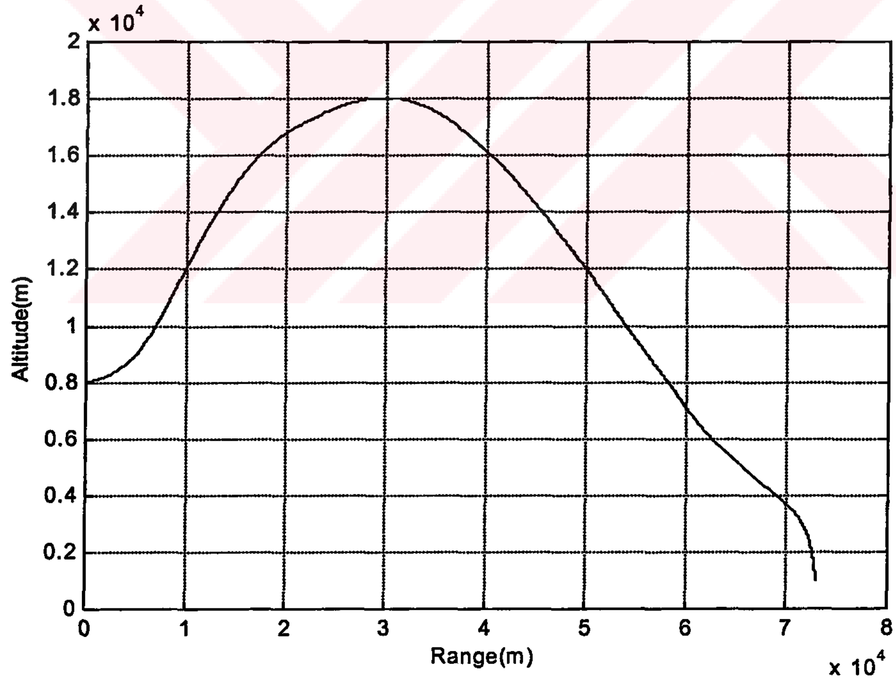


Figure 4.22 Minimum Time, Altitude versus Range Graph

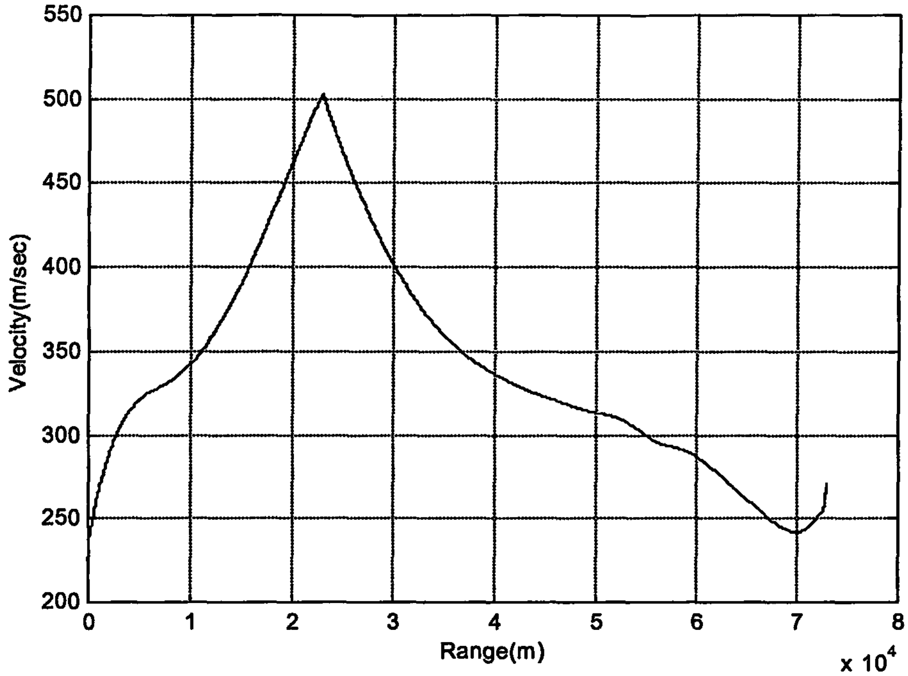


Figure 4.23 Minimum Time, Velocity versus Range Graph

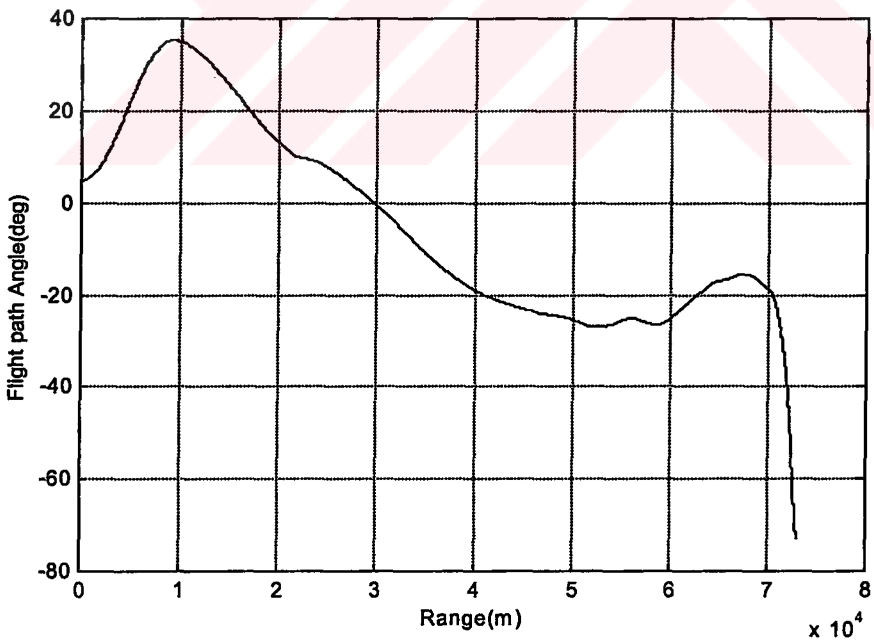


Figure 4.24 Minimum Time, Flight Path Angle versus Range Graph

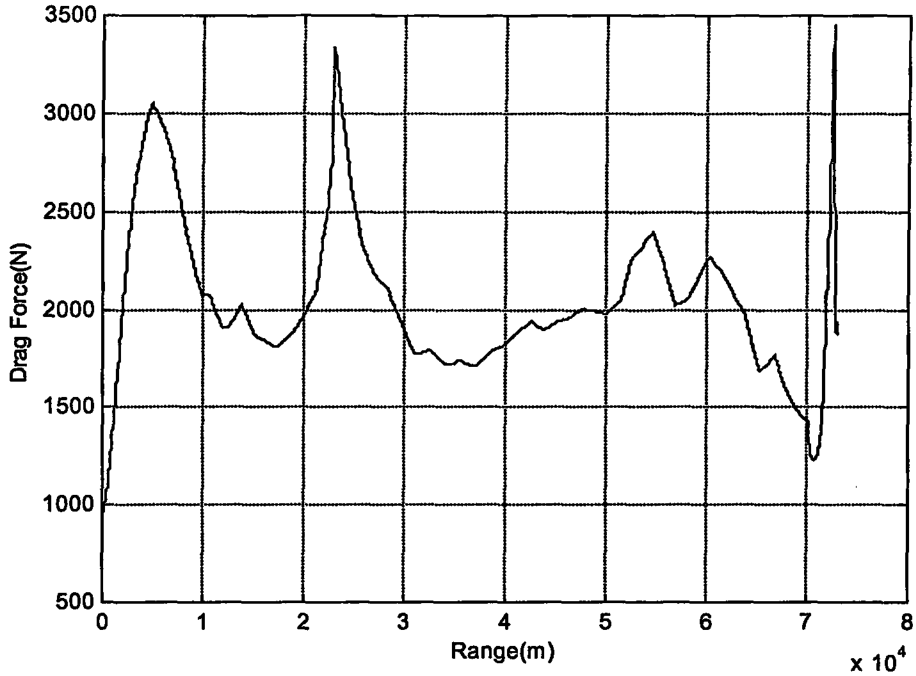


Figure 4.25 Minimum Time, Drag Force versus Range Graph

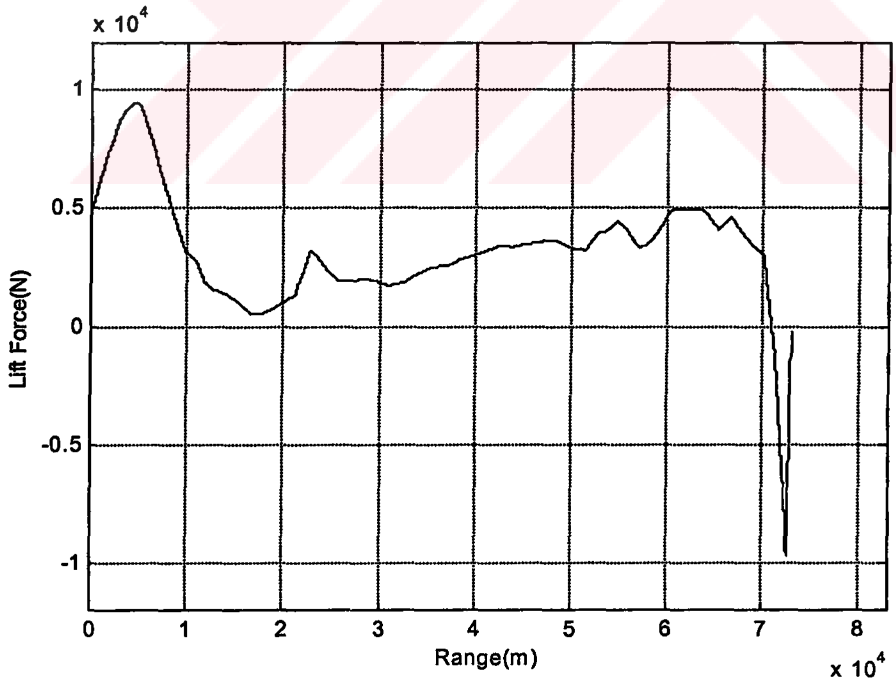


Figure 4.26 Minimum Time, Lift Force versus Range Graph

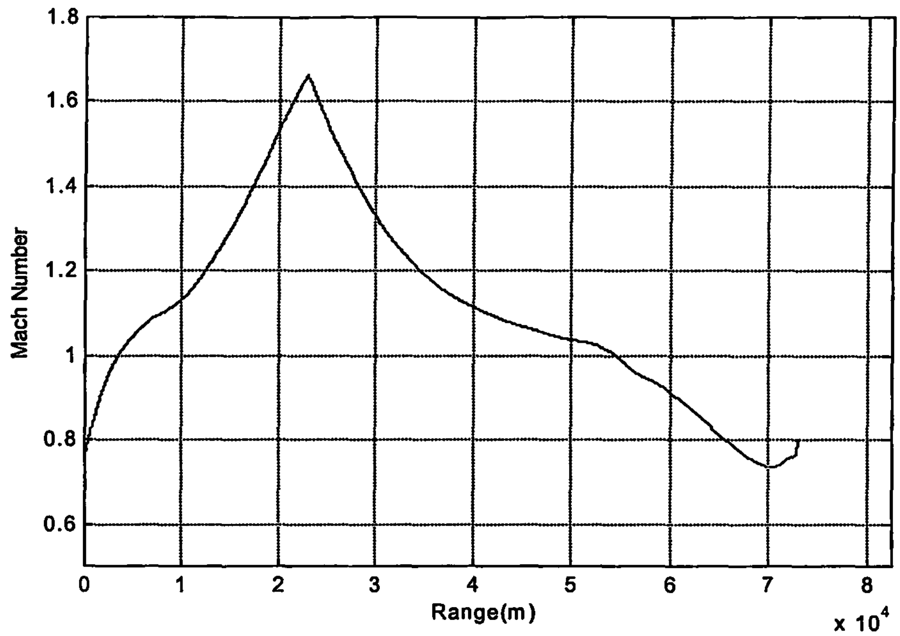


Figure 4.27 Minimum Time, Mach Number versus Range Graph

4.3 Comparison of Flight Time to the Energy Approach

There are two approaches in trajectory optimization. Equal energy approach and equal time approach [11, 13]. In the equal time approach, flight time after burnout is an optimization parameter. The nodes are equally spaced in time before and after burnout. In the equal energy approach, nodes are again equally spaced in time before burnout. However after burnout the total energy (kinetic and the potential energy) of the missile is continuously monitored and the equations are integrated until the final energy specified by terminal constraints is attained. Müge [13] used the time and energy approaches together with the Hide-and-Seek simulated annealing algorithm and showed that energy approach is superior to the time approach.

The purpose of this section is to carry out a comparison using the genetic heuristics. In order to start the algorithm you need an initial population. This population is chosen randomly. To have a meaningful comparison five different populations named SET 1, SET 2, SET 3, SET 4 and SET 5 are chosen randomly where each population has 50 members (chromosomes).

To carry out a comparison each run is terminated just after the cost and range values attained a value higher than 73 km. These results are listed in Table 4-10 and in Table 4-11. In both studies same initial population sets are used. Each member had 60 angle of attack parameter. In addition, time approach had flight time as an additional optimization parameter. Average results are given in Table 4-12. It is seen that energy approach is superior to the flight time in all cases. So in this thesis, energy approach is used.

Table 4-10 Energy Approach Optimization Results for 5 Different Initial Population Sets

	SET 1	SET 2	SET 3	SET 4	SET 5
Generation number	84	40	89	204	34
Crossover rate	08. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50
Mutation Rate	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01
Population size	50	50	50	50	50
Range (m)	77809	74675	74951	73287	75060
Height (m)	1012	1002	997	1003	997
Velocity (m/sec)	270.7	270.8	270.8	270.9	270.9
Flight path angle (deg)	-73.1	-74	-73.5	-75	-74
Fitness Value	74667	74581	74778	73014	74911

Table 4-11 Time Approach Optimization Results for 5 Different Initial Population Sets

	SET 1	SET 2	SET 3	SET 4	SET 5
Generation number	1836	424	2006	901	2015
Crossover rate	0.8. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50	0.8. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50
Mutation Rate	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01
Population size	50	50	50	50	50
Range (m)	74921	76482	73430	75794	74199
Height (m)	1002	989	1001	1008	993
Velocity (m/sec)	272.1	270.6	269.9	272.2	273
Flight path angle (deg)	-74	-73.1	-73.8	-73.3	-75.7
Fitness Value	74791	74305	73362	74271	73317

Table 4-12 Average Value of the Results Given in Tables 4-11 and 4-12

	Energy Approach	Time Approach
Generation number	71	1581
Crossover rate	08. 0.75 0.70 0.65 0.50	08. 0.75 0.70 0.65 0.50
Mutation Rate	0.1 0.04 0.02 0.015 0.01	0.1 0.04 0.02 0.015 0.01
Population size	50	50
Range (m)	75812	74715
Height (m)	1004	1004
Velocity (m/sec)	270.8	271.4
Flight path angle (deg)	-73.	-73.7
Fitness Value	74675.717	74141.477

4.4 Effects of the Algorithm Parameters on the Optimization Results

4.4.1 The Effect of Population Size

In genetic algorithms, population size should be determined properly. If there are only a few members in a population, GA has a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many members, GA slows down. The best population size and the generation size are application dependent. In Table 4-13, Table 4-14 and Table 4-15, optimization algorithm for maximum range problem is run starting from 5 different initial populations; tables are formed using the same set of initial populations while the population sizes differ for each table being 25, 50 and 100 respectively. The runs are terminated just after the cost and range values attained a value higher than 73 km. Average values are given in Table 4-16. It is seen that, population size with 25 members is not enough for diversity, it needs more crossover and mutation to increase it and to reach the defined fitness value. Population size with 100 members needs minimum number of generation among them but does not alter the program too much and slows down it. It seems that, the population size with 50 members is sufficient for this optimization problem.

In Figure 4.28, altitude with respect to range graph for 5 different initial population tabulated in Figure 4.15, are plotted. It should be noted that although the range attained by these trajectories are quite close, there is substantial difference between the trajectories, for example between the maximum altitudes achieved. This is due to the very nonlinear nature of the trajectory optimization problem.

Table 4-13 Results for Population with 25 Members for 5 Different Initial Population

	SET 11	SET 12	SET 13	SET 14	SET 15
Generation number	83	399	44	225	107
Crossover rate	0.80	0.80	0.80	0.80	0.80
Mutation Rate	0.10	0.10	0.10	0.10	0.10
Population size	25	25	25	25	25
Final Range (m)	76523	73676	77018	77644	78942
Final Height (m)	1004	99	990	1010	999
Final Velocity (m/sec)	270.6	271.2	271.4	270.4	271.1
Final flight path angle (deg)	-74.7	-74.5	-74.4	-74.4	-75.2
Fitness Value	75534	73239	74619	73606	78881

Table 4-14 Results for Population with 50 Members for 5 Different Initial Population

	SET 21	SET 22	SET 23	SET 24	SET 25
Generation number	177	67	132	33	83
Crossover rate	0.80	0.80	0.80	0.80	0.80
Mutation Rate	0.10	0.10	0.10	0.10	0.10
Population size	50	50	50	50	50
Final Range (m)	77555	76174	77774	76047	76410
Final Height (m)	998	990	997	1004	1001
Final Velocity (m/sec)	271	271.2	271.1	270.8	270.9
Final flight path angle (deg)	-74.70	-74.4	-74	-75.6	-73.8
Fitness Value	77363	73754	75657	74804	73546

Table 4-15 Results for Population with 100 Members for 5 Different Initial Population

	SET 31	SET 32	SET 33	SET 34	SET 35
Generation number	55	51	215	151	43
Crossover rate	0.80	0.80	0.80	0.80	0.80
Mutation Rate	0.10	0.10	0.10	0.10	0.10
Population size	100	100	100	100	100
Range (m)	75232	76121	74384	77501	78970
Height (m)	1000	1001	993	1001	1000
Velocity (m/sec)	270.9	270.9	271.2	270.7	270.8
Finalflight path angle (deg)	-74.7	-74.5	-74.5	-73.6	-75.9
Fitness Value	75108	75467	73010	73370	77190

Table 4-16 Comparison of Population Sizes w.r.t Generation Number

Popsize	25	50	100
Generation number	138	94	85
Crossover rate	0.80	0.80	0.80
Mutation Rate	0.10	0.10	0.10
Population size	25	50	100
Fitness Value	76007	74319	74649

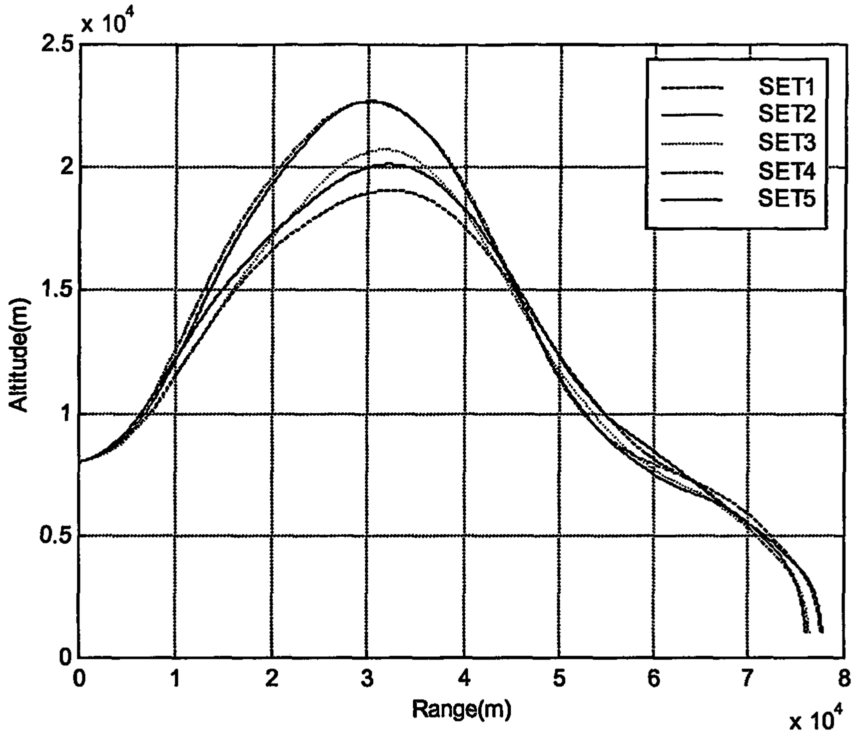


Figure 4.28 Maximum Range, Altitude vs. Range Graph for 5 Different Initial Population Sets with 50 Members

4.4.2 Comparison of Number of Genes in Each Chromosome

Number of genes in each chromosome of the population is the other important parameter in genetic algorithms. To test the effect of the number of genes for the convergence of the maximum range optimization algorithm, 30, 45, 60 and 75 angle of attack values given in Table 4-17 are used. Again 5 different initial populations are used for this problem. The optimization is stopped as soon as the cost and range achieved a value higher than 73 km. The average results are given in Table 4-18. It is seen that 30 genes are not enough for the algorithm to converge to the desired range in a few generations. 45 genes give better results than 30 genes.

As the number of genes is increased to 60, even better results are obtained. However the results obtained using 75 genes are worse compared to 45 and 60 genes. This study shows that more genes are better for trajectory optimization problems up to a point where the algorithm gets too slow to handle the increasing number of parameters.

Table 4-17 Effect of Number of Genes on the Maximum Range Optimization Problem

		Final Fitness	Final Range	NumberofGeneration
30 genes	SET 1	73039	73042	858
	SET 2	73159	73159	393
	SET 3	73054	73181	663
	SET 4	73057	73284	1124
	SET 5	73021	73087	798
45 genes	SET 1	73004	74181	485
	SET 2	73464	73558	67
	SET 3	73570	74702	222
	SET 4	73282	73321	183
	SET 5	73971	74236	59
60 genes	SET 1	73507	78831	59
	SET 2	78566	78579	119
	SET 3	73110	73129	54
	SET 4	74437	78449	87
	SET 5	74779	74951	89
75 genes	SET 1	73365	73366	297
	SET 2	73022	73100	684
	SET 3	73327	73327	395
	SET 4	29812	29872	1000
	SET 5	73225	73276	446

Table 4-18 Effect of Number of Genes on the Maximum Range Optimization Problem

	Final Fitness	Number of Generation
30 Genes	73081	779
45 Genes	73439	157
60 Genes	74241	78
75 genes	73164	508

In Hide and Seek simulated annealing algorithm [13], trajectory optimization problem was run using 30 and 60 angle of attack parameters. The optimization was stopped when the cost and the range achieved a value higher than 70 km. In order to compare the difference between the SA and GA, optimization program is rerun and the results are tabulated in Table 4-19. The number of function of evaluation required by SA is less than the number of function of evaluation required by GA.

Table 4-19 Average Number of Function of Evaluation Required by GA & SA in Maximum Range Problem

	SA Results		GA Results	
	Final Cost	Average # of func. of evakuation	Final Fitness	Average # of func. of evakuation(generation)
30 parameters	70182.52	2134	70547.85	14300 (286 gen.)
60 parameters	70180.2	686	73388.4	3550 (71 gen.)

4.4.3 Crossover Probability

Crossover probability shows how frequently the crossover is applied during the algorithm. If it is low, more generations will be needed to reach the optimum point .If it is high, the genetic code will not be transferred to the next generation and will become extinct. In Table 4-20, 5 different crossover probabilities are applied to the 5 different initial population sets where each set contained 50 members with 60 genes. The problem solved is again the maximum range trajectory optimization problem. The algorithm is stopped as soon as the cost and range achieve a value higher than 80 km. Note that this range is higher than those selected in the previous case studies. This is because the results obtained at lower ranges were not sufficiently discriminatory. Maximum 500 generations are used during the algorithm. From Table 4-22, it is seen that when the crossover probability increases, fewer generations are needed to reach the desired range. Among them, the most effective result is obtained using variable crossover which is decreasing from 80% to 50%.

4.4.4 Mutation Probability

Mutation probability shows how frequently the mutation is applied during the algorithm. In Table 4-21, 5 different mutation probabilities are tested using 5 different initial population sets. The objective is again to achieve the maximum range. The algorithm is stopped as soon as the cost and range exceeds 80 km. In general, if the mutation probability is too small, it does not increase the diversity in the population, if it is too high some useful genes are lost. Same is obtained in this study for low (2%) and high (25%) mutation values. More generations are needed to achieve the optimum result. In most cases the algorithm terminated after preset 500 generations without giving an optimum. For average mutation value of 10% all sets except one achieved the desired optimum. It is also decided to check what happens when variable mutation probability is used. For this purpose first 100 generations were subjected to a mutation probability of 10% then it is decreased to 4%, 2%, 1.5% and 1% respectively for each 100 generation. Note that in all cases the optimum is achieved within 200 generations thus mutation probability of less than 4% is never used. The reverse is also applied starting with mutation probability of 1%. In this case the results were worse than the previous ones. Three out of 5 population terminated without achieving the desired range. Among them, the most effective result is obtained using variable mutation which is decreasing from 10% to 1%.

Table 4-20 Effect of Crossover Probability to Achieve the Desired Range

Crossover Probability		Final Fitness	Final Range	Number of Gener.
20%	SET 1	79906	79906	500
	SET 2	80026	80039	392
	SET 3	80006	80044	370
	SET 4	80124	80126	216
	SET 5	80120	80124	194
50%	SET 1	80026	80026	170
	SET 2	79414	79436	500
	SET 3	80029	80029	471
	SET 4	80183	80311	107
	SET 5	80003	80231	202
80%	SET 1	80158	80352	230
	SET 2	80048	80169	199
	SET 3	80029	80031	340
	SET 4	80222	80253	67
	SET 5	80086	80622	130
80% (0-100 generation)	SET 1	80566	80805	160
75% (100-200 generation)	SET 2	80173	80230	139
70% (200-300 generation)	SET 3	80005	80019	434
65% (300–400 generation)	SET 4	80222	80253	67
50% (400-500 generation)	SET 5	80005	80022	153
10% (0-100 generation)	SET 1	80028	80220	149
20% (100-200 generation)	SET 2	80033	80117	222
30% (200-300 generation)	SET 3	80144	80144	96
40% (300–400 generation)	SET 4	80001	80004	326
50% (400-500 generation)	SET 5	80056	80096	84

Table 4-21 Effect of Mutation Probability to Achieve the Desired Range

Mutation Probability		Final Fitness	Final Range	Number of Gener.
2%	SET 1	80049	80075	349
	SET 2	80315	80339	49
	SET 3	76330	76333	500
	SET 4	79729	79733	500
	SET 5	33322	33323	500
10%	SET 1	80672	80793	139
	SET 2	80384	80964	199
	SET 3	78928	78675	500
	SET 4	80222	80254	67
	SET 5	80156	80325	255
25%	SET 1	71884	71925	500
	SET 2	72537	73867	500
	SET 3	71243	72877	500
	SET 4	73800	74082	500
	SET 5	72485	73549	500
10% (0-100 generation)	SET 1	80566	80805	160
4% (100-200 generation)	SET 2	80173	80230	139
2% (200-300 generation)	SET 3	80005	80019	434
1.5% (300--400 generation)	SET 4	80222	80254	67
1% (400-500 generation)	SET 5	80005	80022	153
1% (0-100 generation)	SET 1	80002	80012	321
1.5% (100-200 generation)	SET 2	78294	76760	500
2% (200-300 generation)	SET 3	75006	75006	500
4% (300--400 generation)	SET 4	77926	77934	500
10% (400-500 generation)	SET 5	80042	80087	119

Table 4-22 Average Values of the Crossover Study

	Final Fitness	Number of Generation
20%	80052	326
50%	80019	281
80%	80097	186
80%75%70%65%50%	80248	150
10%20%30%40%50%	80068	156

Table 4-23 Average Values of the Mutation Study

	Final Fitness	Number of Generation
2%	78703	450
10%	8040	198
25%	72302	500
10% 4% 2% 1.5% 1%	80248	151
1% 1.5% 2% 4% 10%	78741	440

4.5 Perturbation Analysis on Missile Trajectory Optimization Results

For both maximum range and minimum time optimization problems, in time approach, angle of attack values and the flight time after burnout are the input parameters whereas, in energy approach, only the angle of attack values are the input parameters. So the angle of attack values directly affect the success of the missile trajectory. The success of the trajectory then naturally depends on the control logic and the accuracy of the sensors used to guide the missile on the desired trajectory. Then sensitivity of the missile trajectory to these angle of attack values shall be evaluated. For this purpose, a random +/- 0.5 degrees perturbation is applied to the angle of attack values of the maximum range trajectories obtained by equal time and equal energy interpolation simulations. In the time approach flight time after burnout is not perturbed. Results are tabulated in Table 4-24 and Table 4-25.

For both energy and time approaches, the angle of attack history with respect to flight time are presented in Figure 4.29 and Figure 4.31 and the altitude vs. range graphs are given in Figure 4.30 and Figure 4.32. It is seen that the final velocity, flight path angle and angle of attack values obtained by perturbation applied in energy approach is much closer to the original values whereas the range performance of the perturbed time approach results is even better than the original one.

Table 4-24 Results of Perturbed and Original A.O.A Values in Time Approach

	Original A.O.A values	Perturbed A.O.A values
Final Range (m)	80569	80707
Final Height (m)	1000	980
Final Velocity (m/sec)	271.16	266.42
Final flight path angle (deg)	-74.9	-73.4
Final A.O.A (deg)	0.0	-1.04
Total flight time (sec)	285.2	284.5

Table 4-25 Results of Perturbed and Original A.O.A Values in Energy Approach

	Original A.O.A values	Perturbed A.O.A values
Final Range (m)	81582	81111
Final Height (m)	999	1025
Final Velocity (m/sec)	271.1	270.2
Final flight path angle (deg)	-74.9	-73.9
Final A.O.A (deg)	-0.08	-0.09
Total flight time (sec)	297.5	294.0

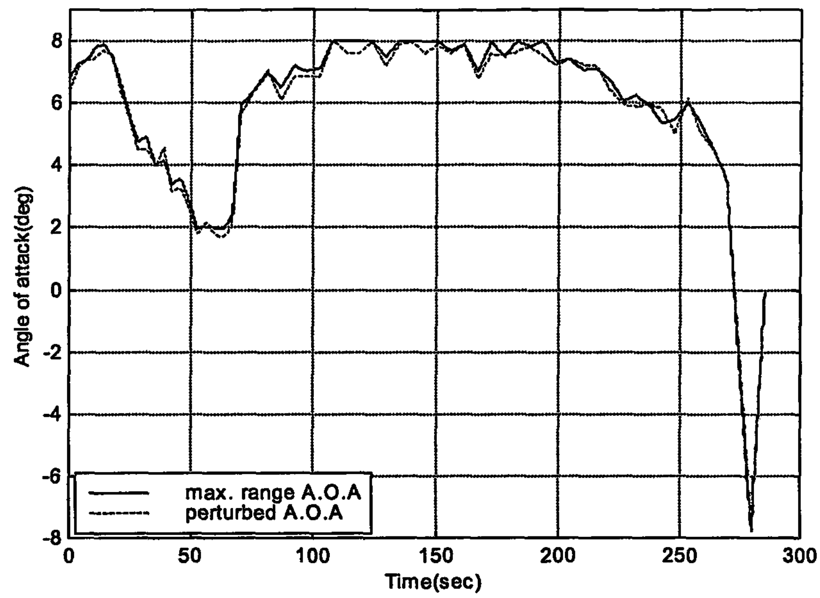


Figure 4.29 Comparison of Perturbed and Original A.O.A Values for the Time Approach

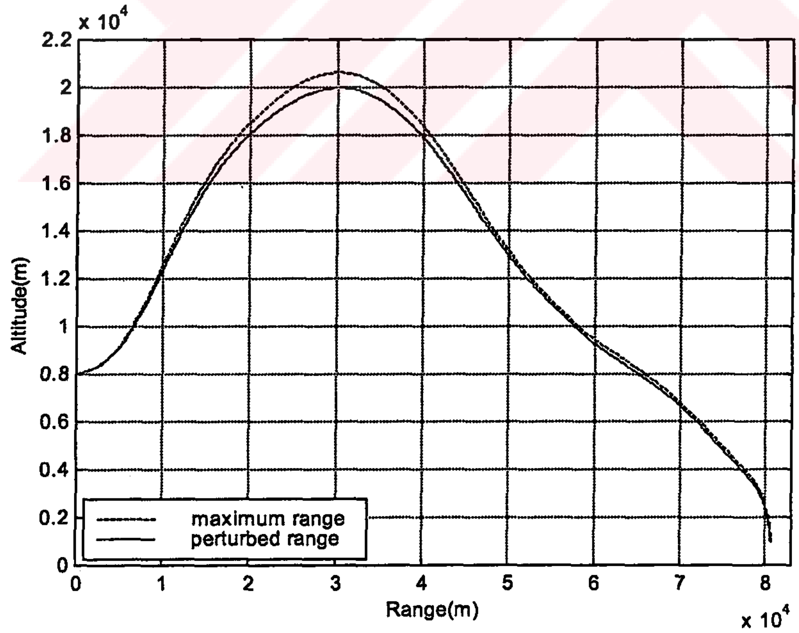


Figure 4.30 Comparison of the Altitude vs. Range Graphs Obtained Using Perturbed and Original A.O.A Values for the Time Approach

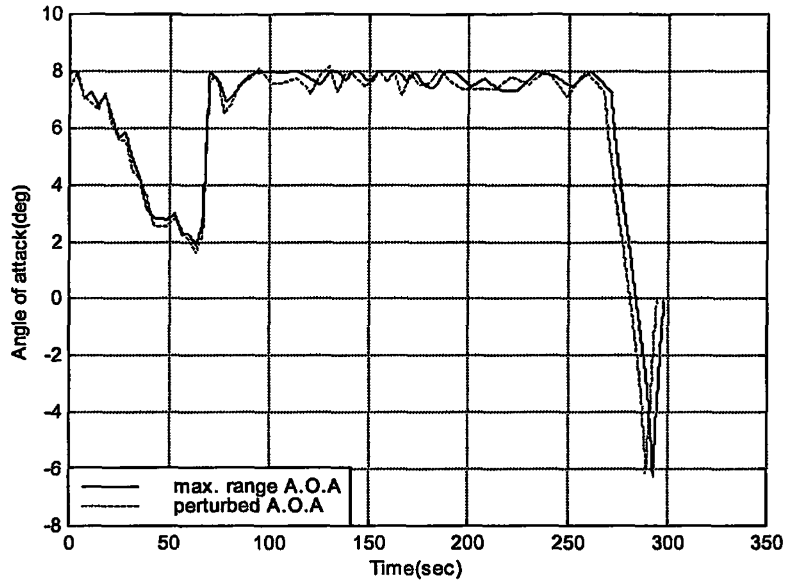


Figure 4.31 Comparison of Perturbed and Original A.O.A Values for the Energy Approach

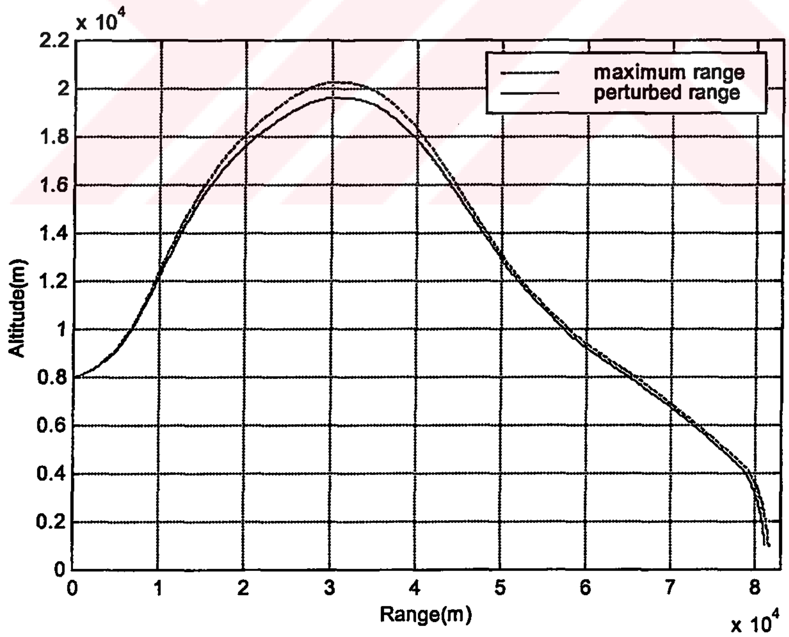


Figure 4.32 Comparison of the Altitude vs. Range Graphs Obtained Using Perturbed and Original A.O.A Values for the Energy Approach

CHAPTER 5

CONCLUSION

In this thesis, trajectory optimization of a highly nonlinear air to surface missiles is investigated using genetic algorithms. Genetic algorithm does not require derivative information and is suitable to find the extrema of highly nonlinear and noncontinuous functions.

For this purpose a two degrees of freedom trim flight mechanics model is employed. Aerodynamic data is calculated using Missile DATCOM. The aerodynamic coefficients for fore X_{cg} location, aft X_{cg} location and after burnout for different angle of attack and Mach number values are tabulated and used during flight simulations.

For a given hypothetical missile two problems, namely, maximum range and specified range minimum flight time problems are considered.

First problem is a maximum range trajectory optimization problem for an air to surface missile. There are 50 members in each population and the evolution is continued for 1000 generations. Angle of attack values are the only control parameters of the optimization problem. For 60 angle of attack values, maximum range obtained is 81582m but for 30 angle of attack values only 75654 m is reached. In the previous study the maximum range achieved by 30 angle of attack parameters was only 73250m [13]. Thus, the range obtained by genetic algorithm is higher than the range obtained by simulated annealing. Specified range, minimum flight time problem is also solved.

In the previous study [13], two approaches in selecting the nodes after burnout where angle of attack parameter is specified was tested: equal energy consumption intervals approach and equal time intervals approach. The named study concluded that equal energy consumption intervals approach is superior to the equal time intervals approach. In this study same problem is solved using genetic algorithm. The energy approach converged to the optimum after 71 generations (3550 function of evaluations) whereas equal time approach required 1581 generations (79050 function of evaluations) to achieve the same range. Thus, equal energy consumption intervals approach is better suited for trajectory optimization problems.

To test the effect of the algorithm parameters on the optimization results, different population sizes, number of genes, crossover probabilities and mutation probabilities are used to solve the maximum range optimization problem. Among them, population size with 50 members with 60 genes (angle of attack values) gave the best results. It is shown that variable crossover probability of 80% to 50% gave better results than fixed crossover probability and a decreasing mutation probability from 10% to 1% helped to achieve the optimum range with the fewer number of generations.

A random +/- 0.5 degrees perturbation is applied to the angle of attack values of the maximum range trajectories obtained by equal time and equal energy interpolation simulations. The results suggest that the optimum trajectories are not much sensitive to small variations on the angle of attack values.

General conclusions of this study can be summarized as follows;

- Defining nodes as equal energy consumption intervals after burnout gives better results than using equal time intervals.
- Genetic algorithm required many more function evaluations than simulated annealing algorithm. However, genetic algorithm achieved a higher range than the simulated annealing algorithm.
- Genetic algorithm has a number of, usually problem specific parameters to be decided. If they are selected properly, the algorithm reaches the global optima in a fewer number of generations.
- This study showed that genetic algorithm is applicable to trajectory optimization problems. In gradient-based algorithms, the computational cost (function evaluation number) increases, usually in a quadratic fashion, with increased number of parameters. However, in genetic algorithm the function of evaluation number does not increase as fast as gradient based algorithms, and it is suitable for problems with many optimization parameters.

Trajectories of many aerospace flight vehicles are important to achieve their missions. Most of the time, the trajectory problems are usually treated separately form the design problem. However, for some problems such as missiles, the combined optimization of design and control variables is very much important. For this reason, the future study should address the combined optimization of missile design and control variables together using genetic algorithm.

REFERENCES

[1] Guevara, J. R., Optimization Strategies for the Synthesis/Design of Highly Coupled, Highly Dynamic Energy Systems, Ph.D. Thesis, Mechanical Engineering Department,Virginia Tech ,Blacksburg VA , September 2000

[2] Anderson, M.B., Burkhalter, J.E. and Jenkins, R.M., "Missile Aerodynamic Shape Optimization Using Genetic Algorithms," AIAA 99-0261, 37th Aerospace Sciences Meeting & Exhibit, January 11-14, 1999, Reno, NV

[3] Tava, M. and Suzuki, S., "Application and Testing of a SQP Code to Some Trajectory Optimization Problems", JSME-OPID, 1997

[4] Arslan, E. M., Optimal Trajectories for Air-to-Surface Missiles Using Direct Collocation and Nonlinear Programming, M.S. Thesis, Aeronautical Engineering Department, METU, Ankara, March 1995

[5] Tekinalp, 0., Arslan, E.M., "Optimal Mid-course Trajectories for Precision Guided Missions with Collocation and Non-Linear Programming," Proceedings of the 20th Congress of the International Council of Aeronautical Sciences, V.1, pp. 593-600, 1996

[6] Tekinalp 0., Arslan E.M., "Multidisciplinary Design Optimization Tool for Air to Surface Missiles," Proceedings of the 3rd World Congress of Structural and Multidisciplinary Design Optimization, 17-21 May, Univ of Buffalo, Amherst, NY, 1999

[7] Haupt R.L. and Haupt S.E, Practical Genetic Algorithms, John Wiley & Sons Inc., 1998

[8] Kirkpatrick, S., et. al., "Optimization by Simulated Annealing", Science, Vol.220, Number 4598, 13 May 1983

[9] Bélisle, C.J.P., Romeijn, H.E., and Smith, R.L., Hide-and-Seek: A Simulated Annealing Algorithm for Global Optimization, Department of Industrial and Operations Engineering, Technical Report 90-25, University of Michigan, Ann Arbor, 1990

[10] Lu, Ping and Khan, Asif, "Nonsmooth Trajectory Optimization: An Approach Using Continuous Simulated Annealing," Journal of Guidance, Control and Dynamics, Vol.17, No.4, July-August 1994

[11] Utalay, S., Trajectory and Multidisciplinary Design Optimization of Missiles Using Simulated Annealing, M.S. Thesis, Aeronautical Engineering Department, METU, Ankara, January 2000

[12] Tekinalp, O., Utalay S., "Simulated Annealing for Missile Trajectory Planning and Multidisciplinary Missile Design Optimization," AIAA 38th Aerospace Sciences Meeting, Reno, NY, January 10-13, 2000

[13] Bingöl, M., Trajectory and Multidisciplinary Design Optimization of Missiles Using Simulated Annealing, M.S. Thesis, Aeronautical Engineering Department, METU, Ankara, January 2000

[14] Oyama,A., Obayashi S., Nakahashi K., "Transonic Wing Optimization Using Genetic Algorithm," Paper No.97-1854, AIAA, 1997

[15] Noton, M., "Orbital Strategies Around a Comet by Means of a Genetic Algorithm," Journal of Guidance, Control, and Dynamics, Vol. 18, No. 5, 1995, pp.1217-1220

[16] Pinon, E. III and Fowler, W., "Lunar Launch Trajectory Optimization Using a Genetic Algorithm," Paper No. 95-142, AAS/AIAA Space Flight Mechanics Meeting, February 1995

[17] Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer., 1996, USA

[18] Macfadzean, Robert, Surface Based Air Defense System Analysis, MacGraw Hill Inc., 1992

[19] Chin, S.S, Missile Configuration Design, MacGraw Hill Inc., 1961

[20] Sutton G.P, Rocket Propulsion Elements, John Wiley & Sons Inc., 1986

[21] Bryson,A.E. and Y.C.Ho, Applied Optimal Control, Waltham, MA.Blasisdell Publishing Co., 1969.(2nd. Ed. Published by Halsted Press, 1975)

[22] Blake, W.B., Missile DATCOM User's Manual, AFRL-VA-WP-TR-1998-3009, Wright Patterson Air Force Base, Ohio USA,Februrary 1997

[23] Obitko M., Genetic Algorithms http:/cs.felk.cvut.cz/xobitko/ga, 1998

APPENDIX A

AIR TO SURFACE MISSILE MODEL

A.1 Missile Aerodynamic Inputs

The generic baseline ASM configuration chosen for this study is shown in Figure A.1. It has an end-burning solid propellant motor, warhead section, guidance

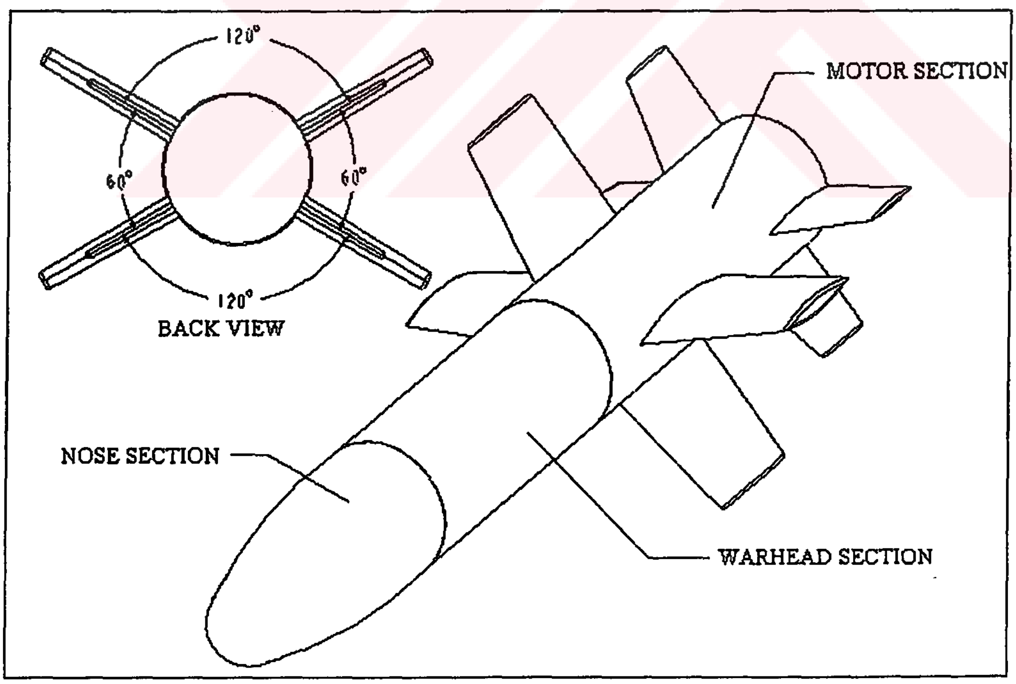


Figure A.1 Generic Air-to-Surface Missile Configuration

section including Inertial Navigation System (INS) and a digital autopilot for high accuracy midcourse flight phase navigation, with a seeker and four electro-mechanically controlled tails and four wings.

A.1.1 Reference Quantities

Main reference parameters of the missile used are illustrated in Figure A.2 and listed in Table A-1.

Table A-1 Input Reference Parameters

Parameter	Value
Reference Length (LREF)	0.45 m
Reference Diameter (DREF)	0.45 m
Position of center of gravity (XCG)	2.1884 m

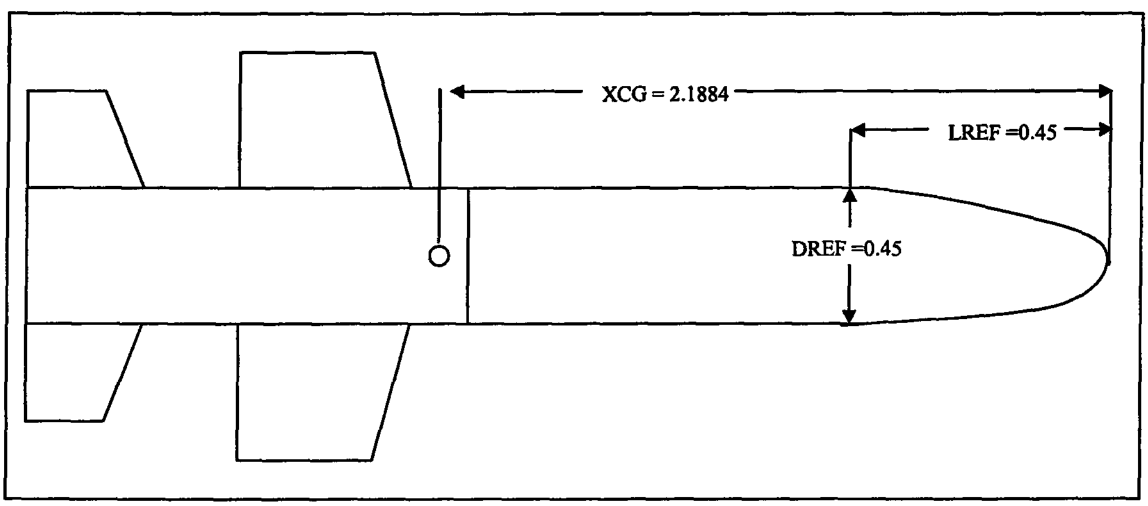


Figure A.2 Missile Reference Quantities

A.1.2 Axisymmetric Body Parameters

Main axisymmetric body parameters of the missile used are illustrated in Figure A.3and listed in Table A-2.

Table A-2 Axisymmetric Body Parameters

Parameter	**Value**
Length of the ogive nose (LNOSE)	1.00 m
Length of the warhead (LWHEAD)	1.10 m
Length of Center (LCENTR)	2.5358 m
Exit Diameter (DEXIT)	0.45 m
Center Diameter (DCENTR)	0.45 m
Nose Diameter (DNOSE)	0.45 m
Nose Bluntness Radius (BNOSE)	0.13 m

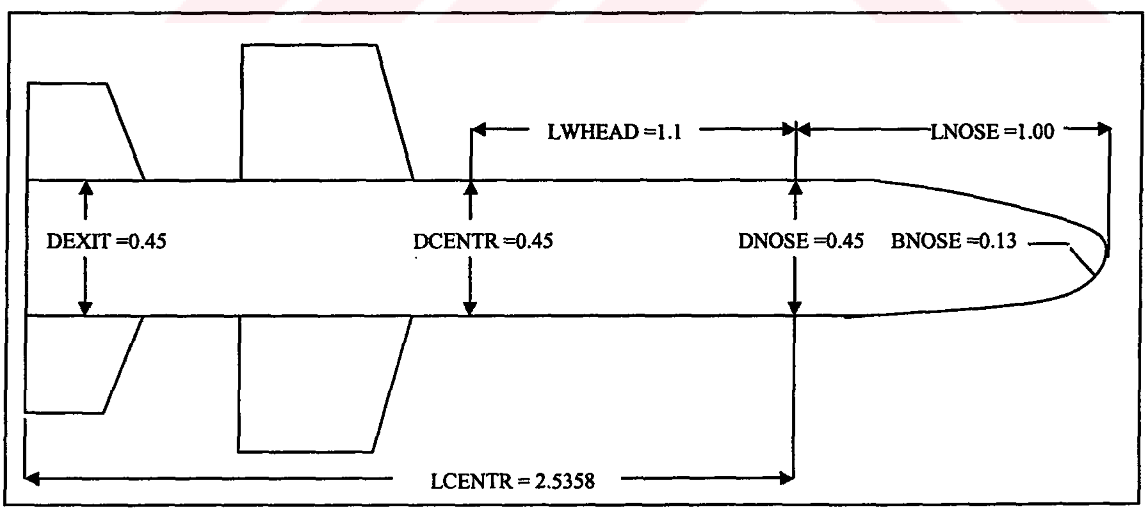


Figure A.3 Missile Axisymmetric Body Parameters

A.1.3 Wing Inputs

There are four electromechanically controlled wings on this missile. Main wing parameters of the missile used are illustrated in Figure A.4. These parameters which are used in the missile wing design are listed in Table A-3.

Table A-3 Missile Wing Parameters

Parameter	Value
Wing semi span length (SSPAN)	0.45 m
Wing chord length (CHORD)	0.4294 m, 0.55 m
Wing leading edge sweep angle ($\Lambda_{LE,wing}$)	15°
Wing trailing edge sweep angle ($\Lambda_{TE,wing}$)	0°
Length of chord leading edge (XLE)	2.5358 m

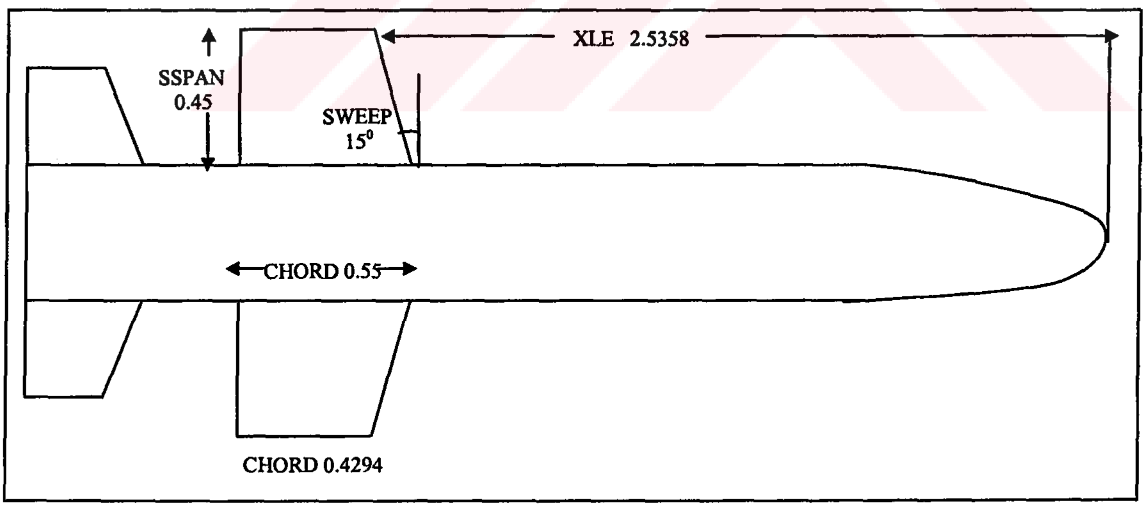


Figure A.4 Missile Wing Parameters

A.1.4 Tail Inputs

There are four electromechanically controlled tails on this missile. Main tail parameters of the missile used are illustrated in Figure A.5. These parameters which are used in the missile tail design are listed in Table A-4.

Table A-4 Missile Tail Parameters

Parameter	Value
Tail semi span length (SSPAN)	0.30 m
Tail chord length (CHORD)	0.260108 m, 0.40 m
Tail leading edge sweep angle ($\Lambda_{LE,tail}$)	25°
Tail trailing edge sweep angle ($\Lambda_{TE,tail}$)	0°
Length of chord leading edge (XLE)	3.1358 m

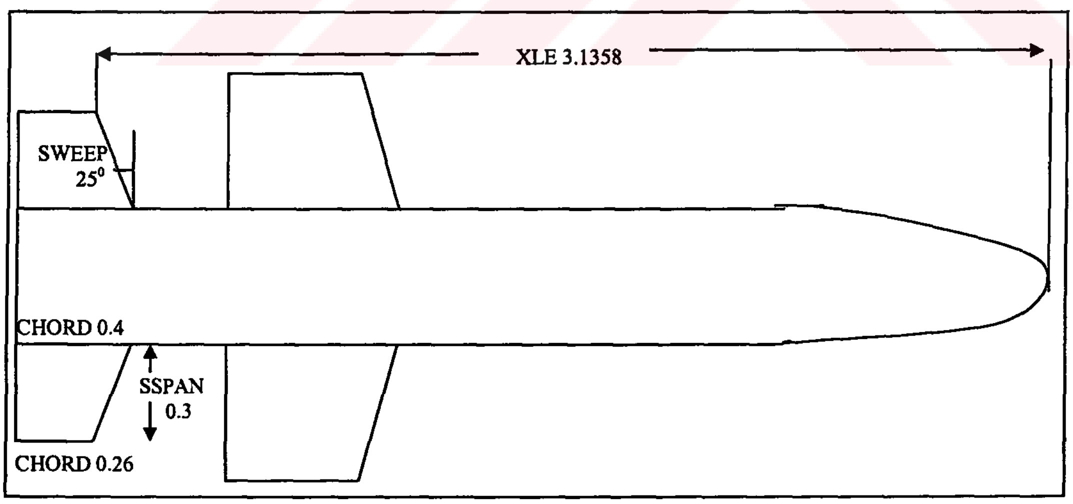


Figure A.5 Missile Tail Parameters

A.1.5 Airfoil Type

Cross-section type used both for wing and tail is symmetric hexagonal as shown in Figure A.6.

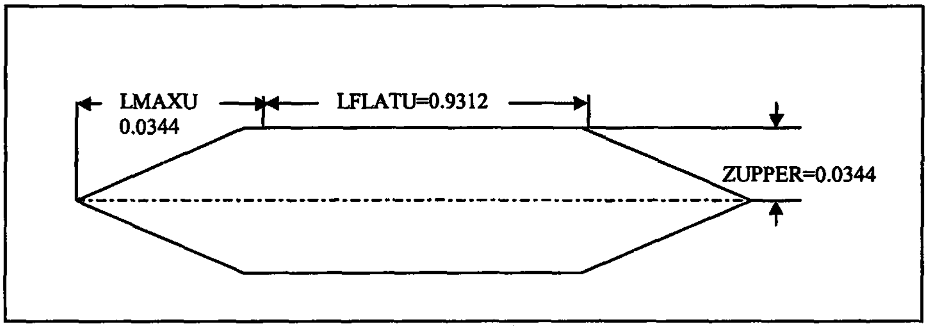


Figure A.6 Airfoil Parameters

A.2 Missile DATCOM Program Capabilities

Time dependent design. and analysis of missiles requires the use of rapid and accurate analytical procedures to determine their aerodynamic characteristics. The fundamental purpose of Missile DATCOM™ [19] is to provide an aerodynamic design tool which has the predictive accuracy suitable for preliminary design, and the capability for the user to easily substitute methods to fit specific applications. The computer code is capable of addressing a wide variety of conventional missile designs. A conventional missile is one, which is comprised of the following:

1. An axisymmetric or elliptically-shaped body
2. One to four fin sets located along the body between the nose and base. Each fin set can be comprised of one to eight identical panels attached around the

body at a common longitudinal position. Each fin may be deflected independently.

3. An airbreathing propulsion system.

The program has the capability to perform a static trim of the configuration, using any fin set for control with fixed incidence on the other sets.

A.2.1 Input Definition

Inputs to the program are grouped by "case". A "case" consists of a set of input cards, which define the flight conditions and geometry to be run. Provisions are made to allow multiple cases to be run. The successive cases can either incorporate the data of the previous case (using input card SAVE) or be a completely new configuration design. The SAVE feature, for example, permits the user to define a body and wing (or canard) configuration in the first case and vary the tail design for subsequent cases. Flexibility has been maintained for all user inputs and outputs like unit system, coordinates, airfoil sections, flight conditions etc. The input file named as "for005.dat" for Missile DATCOM™ used in this work is listed below:

```
**********
*MELTEM TEZ
**********
CASEID MISSILE
DIM M
SOSE
DERIV DEG
**
***REFERENCE QUANTITIES***
$REFQ
 SREF=0.159043,
 LREF=0.450000,
 XCG=2.188400,$
**
***FLIGHT CONDITIONS******
$FLTCON
 NALPHA=17.0,
 ALPHA=-8.0,-7.0,-6.0,-5.0,-4.0,-3.0,-2.0,-1.0, 0.0, 1.0,
 ALPHA(11)= 2.0, 3.0, 4.0, 5.0, 6.0,   7.0, 8.0,
 NMACH=15.0,
 MACH= 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10, 1.15,
 MACH(11)= 1.20, 1.25, 1.30, 1.35, 1.40,
 ALT=10000.0,$
**
***AXISYMMETRIC BODY INPUTS***
$AXIBOD
 TNOSE=OGIVE,
 LNOSE=1.000000,
 DNOSE=0.450000,
 BNOSE=0.13,
 LCENTR=2.535800,
 DCENTR=0.450000,
 DEXIT=0.450000,$
***FINSET-1 (WING) INPUTS****
$FINSET1
 SSPAN=0.0,0.450000,
 CHORD=0.550000,0.429423,
 XLE=2.285800,
 SWEEP=15.000000,
 NPANEL=4.0,
 PHIF=60.,120.,240.,300.,
 ZUPPER=0.01425,0.01425,
 LMAXU=0.0344,0.0344,
 LFLATU=0.9312,0.9312,$
***FINSET-2 (TAIL) INPUTS****
$FINSET2
 SSPAN=0.0,0.300000,
 CHORD=0.400000,0.260108,
 XLE=3.135800,
 SWEEP=25.000000,
 NPANEL=4.0,
 PHIF=60.,120.,240.,300.,
 ZUPPER=0.01425,0.01425,
 LMAXU=0.0344,0.0344,
 LFLATU=0.9312,0.9312,$
**
 ***BOOST INITIAL TRIM CASE***
$TRIM
 SET=2.,
 DELMAX=25.,
 DELMIN=-25.,
 PANL1=.TRUE.,
 PANL2=.TRUE.,
 PANL3=.TRUE.,
 PANL4=.TRUE.,$
SAVE
NEXT CASE
**
****CASE_2 BOOST FINAL TRIM CASE****
CASEID MUGEMISSILE BOOST FINAL
DELETE REFQ
$REFQ
 SREF=0.159043,
 LREF=0.450000,
 XCG=1.782200,$
$TRIM
 SET=2.,
 DELMAX=25.,
 DELMIN=-25.,
 PANL1=.TRUE.,
 PANL2=.TRUE.,
 PANL3=.TRUE.,
 PANL4=.TRUE.,$
SAVE
NEXT CASE
**
****CASE_3 COAST TRIM CASE****
CASEID MUGEMISSILE COAST
DELETE REFQ
DELETE AXIBOD
$REFQ
 SREF=0.159043,
 LREF=0.450000,
 XCG=1.782200,$
$AXIBOD
 TNOSE=OGIVE,
 LNOSE=1.000000,
 DNOSE=0.450000,
 BNOSE=0.13,
 LCENTR=2.535800,
 DCENTR=0.450000,
 DEXIT=0.,$
$TRIM
 SET=2.,
 DELMAX=25.,
 DELMIN=-25.,
 PANL1=.TRUE.,
 PANL2=.TRUE.,
 PANL3=.TRUE.,
 PANL4=.TRUE.,$
SAVE
NEXT CASE
```

A.2.2 Namelist Inputs

The namelist names have been selected mnemonically related to their physical meaning. The namelists can be input in any order. Only those namelists required to execute the case need be entered. Some namelists used available are as follows:

$FLTCON Flight Conditions (Angle of attack, Mach numbers, etc.)

$REFQ Reference quantities (Reference area, length, etc)

$AXIBOD Axisymmetric body definition

$FINSET*n* Fin descriptions by fin set (*n* is the fin set number; 1,2,3 or 4)

$TRIM Trimming information

$INLET Inlet geometry

A.2.2.1 Namelist FLTCON – Flight Conditions

This namelist defines the flight conditions to be run for the case. A "case" is defined as a fixed geometry with variable Mach number and angles of attack.

NALPHA, number of angles of attack to run, **ALPHA**, angle of attack schedule (matching **NALPHA**), **NMACH**, number of Mach number, **MACH**, Mach number schedule (matching **NMACH**) must be defined as variables.

Reynolds number is always required. In our input file, this requirement is satisfied by specifying the altitude using **ALT**, and the speed using **MACH** (Reynolds number is computed using the Standard Atmosphere model).

A.2.2.2 Namelist REFQ – Reference Quantities

Inputs for this namelist are **SREF**, reference area, **LREF**, reference length and **XCG**, position of center of gravity (moment reference center). These variables are user defined. All coefficients are computed according to these variables so they will be used during the simulation.

A.2.2.3 Namelist AXIBOD – Axisymmetric Body Geometry

An axisymmetric body is defined using this namelist. The geometry is divided into nose, center-body, and aft body sections. The shape, overall length, and base diameter for each section are specified. The program automatically calculates the body contour based upon the segment shapes using geometry generator.

LNOSE	Length of the body segment to where the radius first reaches a maximum
DNOSE	The diameter at the first radius maximum
LCENTR	Length of the body segment where the radius is constant
DCENTR	Diameter of the constant radius segment
LAFT	The remaining body length
DAFT	Diameter at the base

Base drag computed is included in the axial force calculations by specifying the exit diameter as zero (DEXIT=0). Otherwise base drag computed for the body geometry will not be included in the final computed axial force calculations and boost phase calculations are done.

TNOSE and **BNOSE** define the nose shape name and nose bluntness radius respectively.

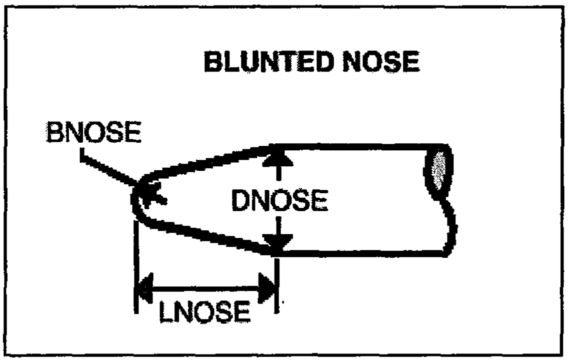


Figure A.7 Nose Model

Note that not all three body sections need to exist on a configuration; for example, a nose-cylinder missile configuration of this thesis does not require definition of an aft body. If **LAFT** is not input, aft body does not exist.

A.2.2.4 Namelist FINSETn – Define Fin Set n

Fin set planform geometry and fin cross-section inputs are described in this part. The user may specify up to four non-overlapping fin sets. The variable "*n*" in the namelist specifies the fin set number. Fin sets must be numbered sequentially from the front to back of the missile beginning with fin set one. This means that FINSET2 must always be aft of FINSET1. The code allows for between 1 and 8 geometrically identical panels to be input per fin set. The panels may be arbitrarily rolled about the body.

Four types of airfoil sections are permitted. Cross-section type used in above input file is symmetric hexagonal as shown in Figure A.8.

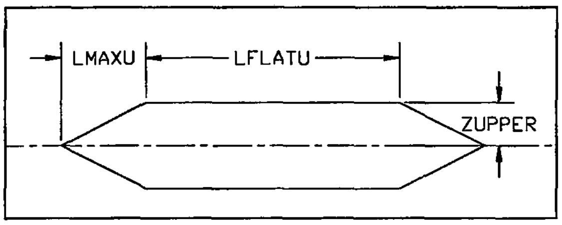


Figure A.8 Fin cross-section Inputs

The user selects "break points" on the panel. A "break point" specifies a change in leading or trailing edge sweep angle as shown in Figure A.9. The location of each "break point" is defined by specifying its semi-span station (**SSPAN**) from the vehicle centerline and distance from the first body station to the chord leading edge (**XLE**).

The panel sweep angle (**SWEEP**) can be specified at any span station for each segment of the panels. If STA=0., the sweep angle input is measured at the segment leading edge; if STA=1., the sweep angle input is measured at the segment trailing edge. It is recommended that exact sweep angles be specified wherever possible; for example, if the panel trailing edge is unswept, specifying SWEEP=0. and STA=1. will minimize calculation error. Then the leading edge sweep will be computed by the code internally using the SSPAN and CHORD inputs.

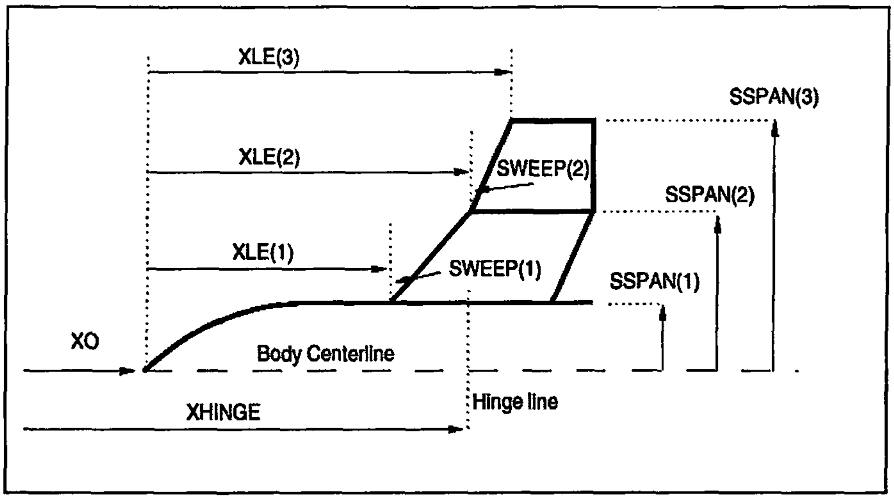


Figure A.9 Selecting Panel Break Point

The number of panels present is defined using the variable **NPANEL**. Each panel may be rolled to an arbitrary position around the body using the variable **PHIF**. **PHIF** is measured clockwise from top vertical center (looking forward from behind the missile) as shown in Figure A.10.

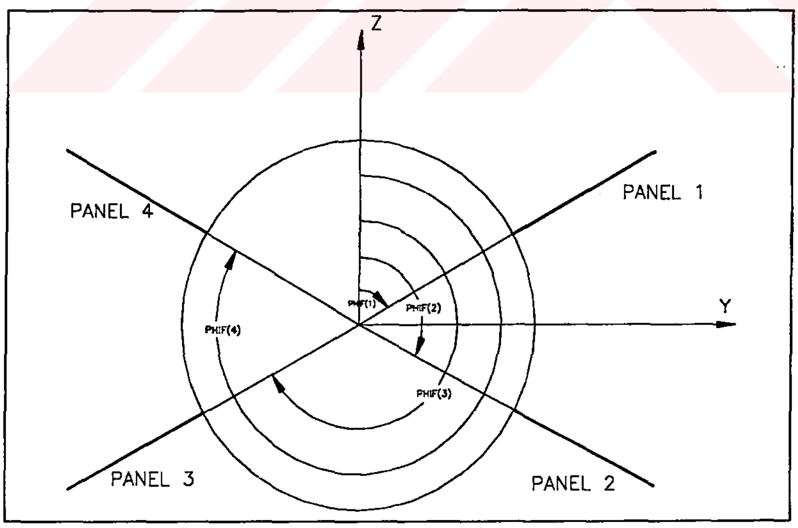


Figure A.10 Fin Numbering and Orientation

A.2.2.5 Namelist TRIM – Trim Aerodynamics

This namelist instructs the program to statically trim the vehicle longitudinally (C_m=0). Only one fin set can be used for trimming. The user only specifies the range of deflection angles desired using **DELMIN** and **DELMAX**; the code will try to trim the vehicle for each angle of attack specified using the allowable fin deflections.

A.2.3 Control Card Inputs

Control cards are one line commands which select program options. Although they are not required inputs, they permit user control over program execution and the types of output desired. Control cards enable the following:

CASEID, * Case titles or comments to the input file and output pages are specified. Up to 73 characters can be specified.

DIM M, DERIV DEG These control cards sets the system of units to meters and All output derivatives to degree.

NEXT CASE This card indicates termination of the case input data and instructs the program to begin case execution. It is required for multiple case "runs:. This card must be the last card input for the case.

SAVE The Save card saves namelist inputs from one case to the following case but not for the entire run. This permits the user to build-up or change a complex configuration, case-to-case, by adding new namelist cards without having to re-input namelist cards of the previous case. When changing a namelist that has been saved, the namelist must first be deleted using the delete control card.

DELETE name1,name2 This control card instruct the program to ignore a previous case namelist input that was retained using the SAVE control card. All previously saved namelists with the names specified will be purged from the input file. Any new inputs of the same namelist will be retained.

SOSE The presence of this control card selects the Second-Order Shock Expansion Method for axisymmetric bodies at supersonic speeds.

A.2.4 Aerodynamic OUTPUT

The program has the capability to perform a static trim of the configuration, using any fin set for control with fixed incidence on the other sets. The code computes the six-component aerodynamics at ten deflection angles for each specified angle of attack, then interpolates for C_m=0. Aerodynamic output available from the trim option are the trimmed aerodynamic coefficients, and trim deflection angle, are output as a function of angle of attack. The nomenclature is as follows:

DELTA	Trim deflection angle
CN	Trimmed Normal force coefficient
CA	Trimmed Axial force coefficient
CL	Trimmed Lift coefficient
CD	Trimmed Drag coefficient
CL/CD	Lift to drag ratio

APPENDIX B

TRAJECTORY OPTIMIZATION RESULTS OF GA & SA

B.1 Maximum Range Trajectory Optimization Results of GA & SA

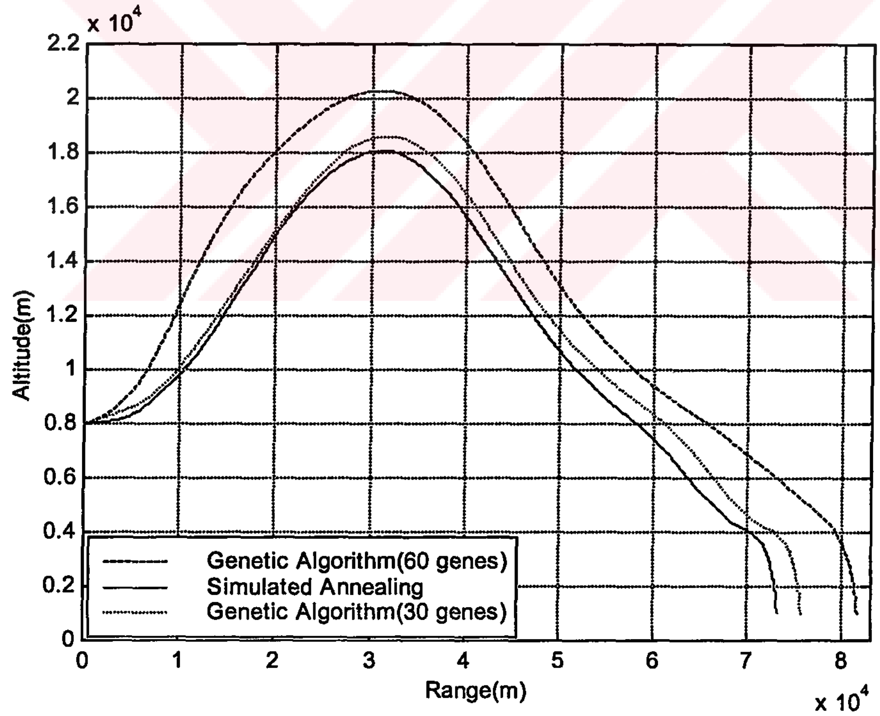


Figure B.1 Maximum range trajectories obtained using GA and SA.

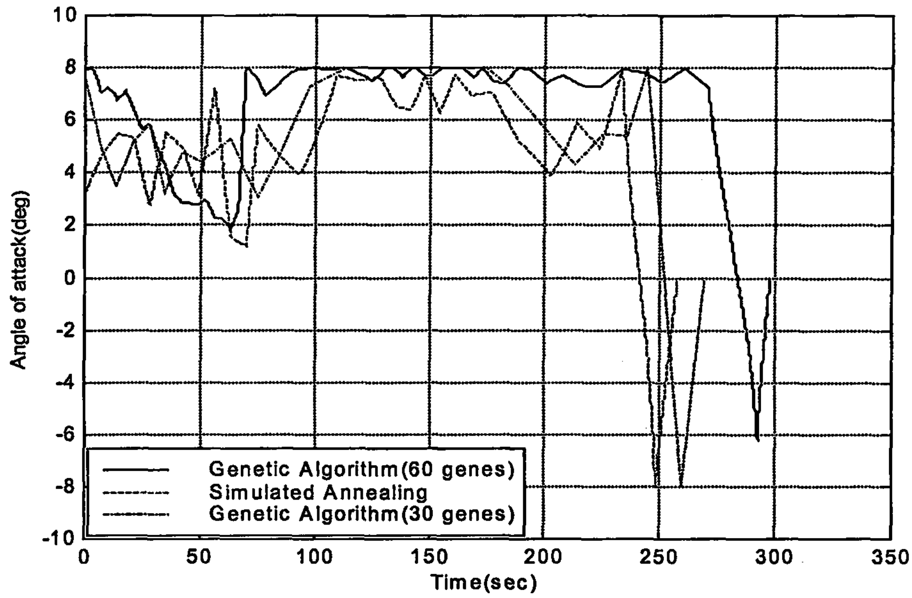


Figure B.2 Angle of Attack History for Maximum Range Trajectory Using GA and SA

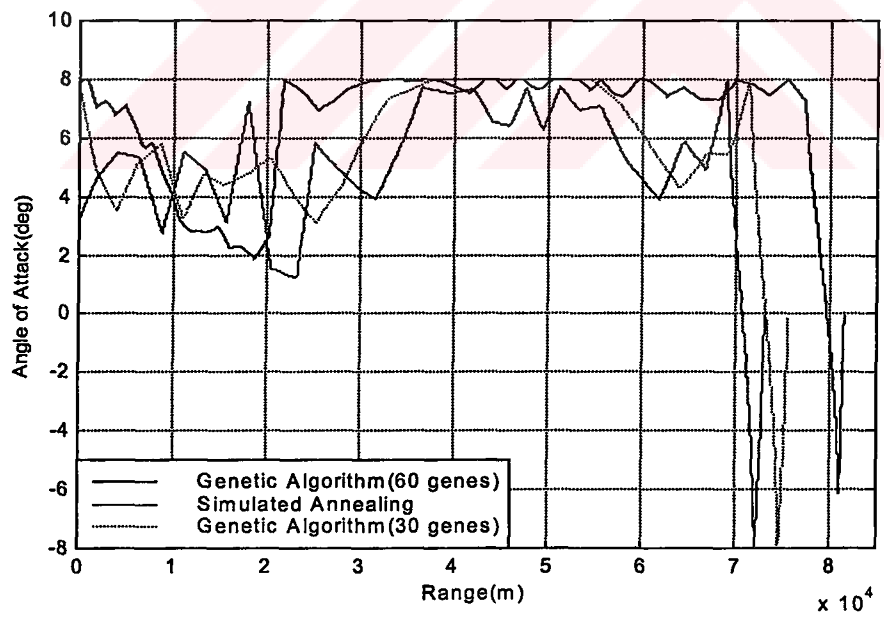


Figure B.3 Maximum Range, Angle of Attack versus Range Graphs Using GA and SA

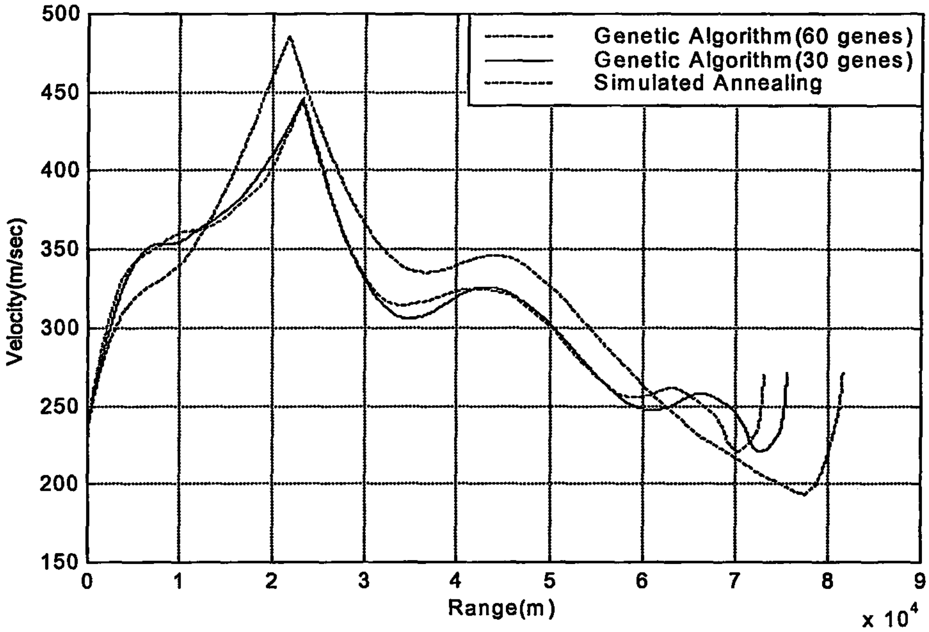


Figure B.4 Maximum Range, Velocity versus Range Graph using GA and SA

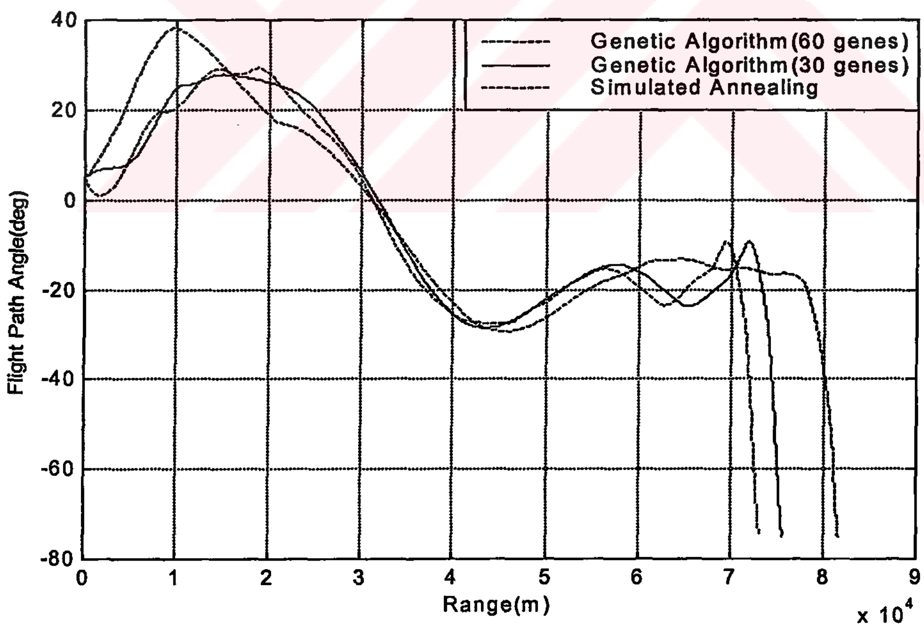


Figure B.5 Maximum Range, Flight Path Angle versus Range Graph using GA and SA

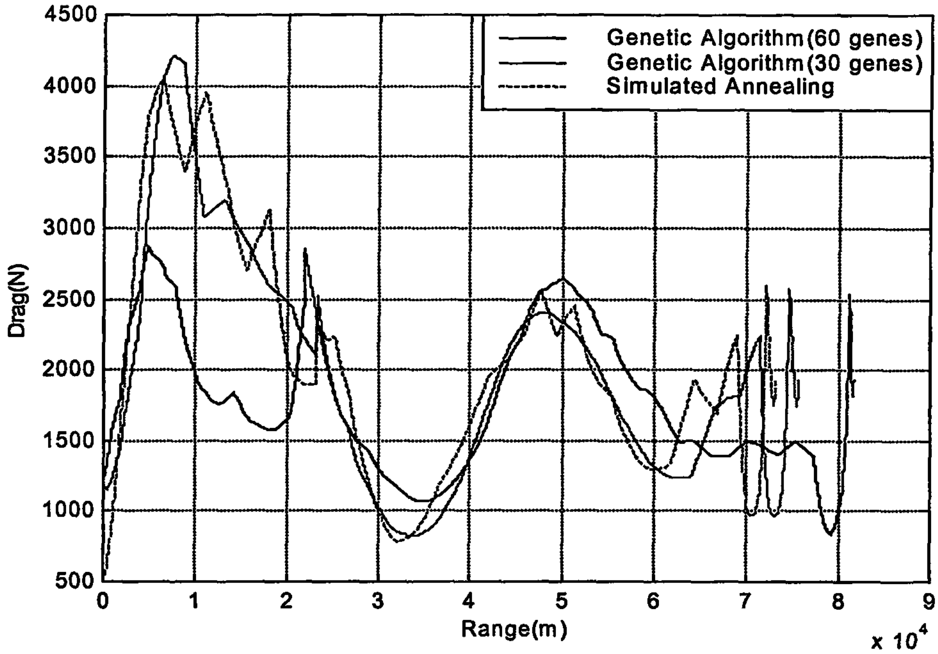


Figure B.6 Maximum Range, Drag versus Range Graph using GA and SA

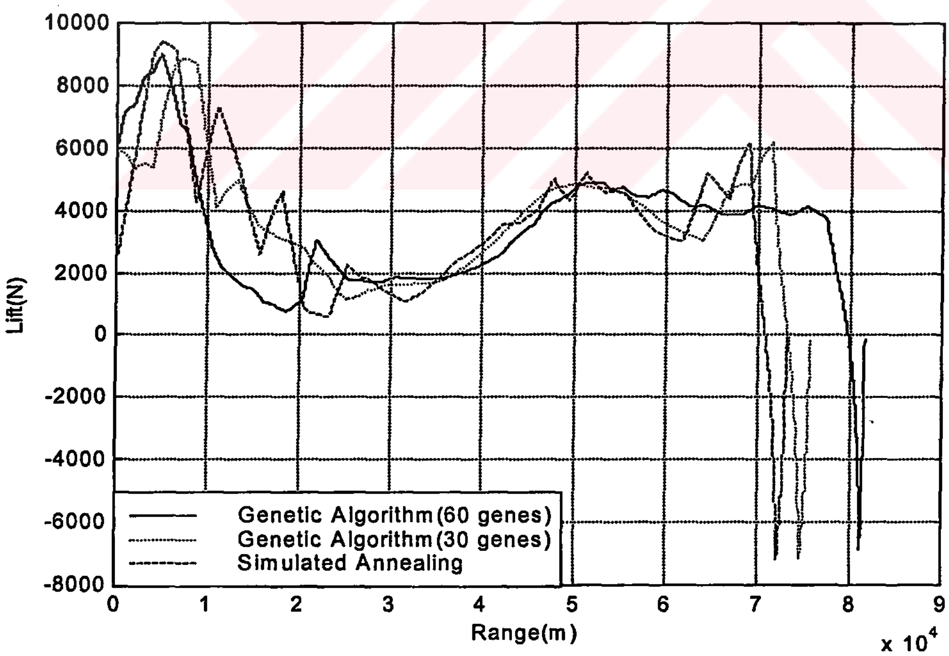


Figure B.7 Maximum Range, Lift versus Range Graph using GA and SA

B.2 Specified Range Minimum Flight Time and Maximum Range Trajectory Optimization Graphs Obtained Using GA

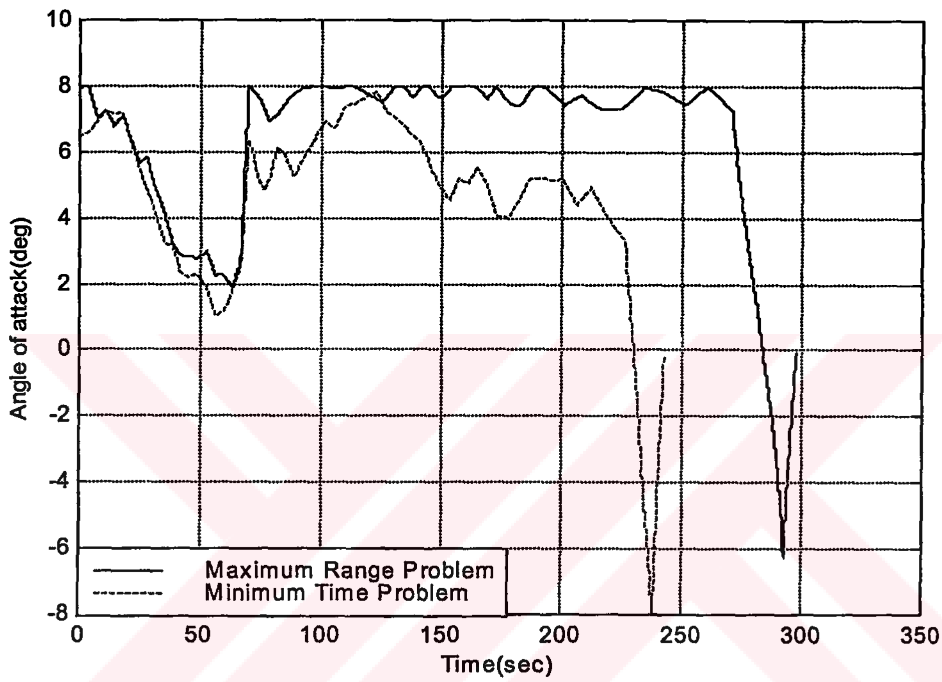


Figure B.8 Angle of Attack -Flight Time Graph for Maximum Range and Minimum Flight Time Trajectories.

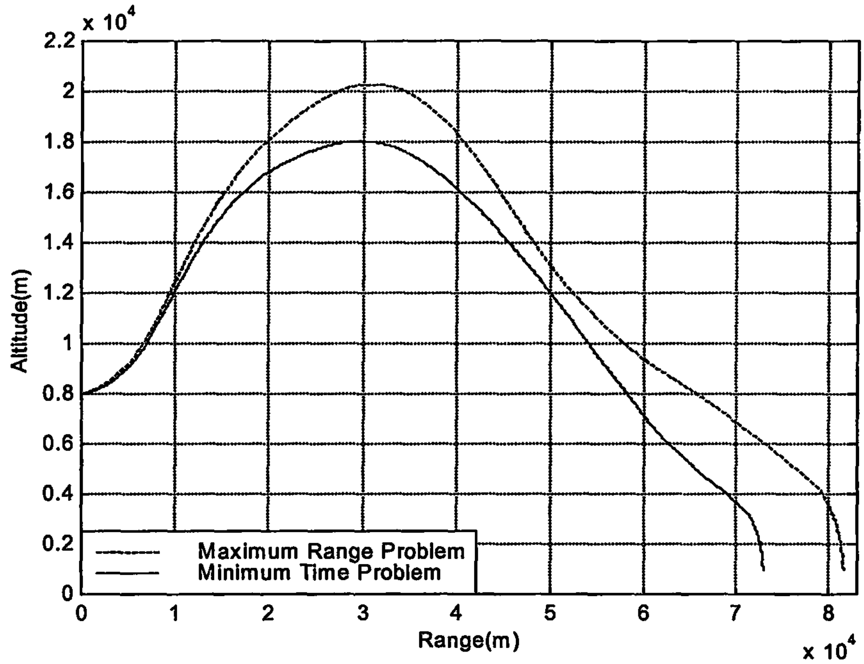


Figure B.9 Altitude-Range Graph for Maximum Range and Minimum Flight Time Trajectories

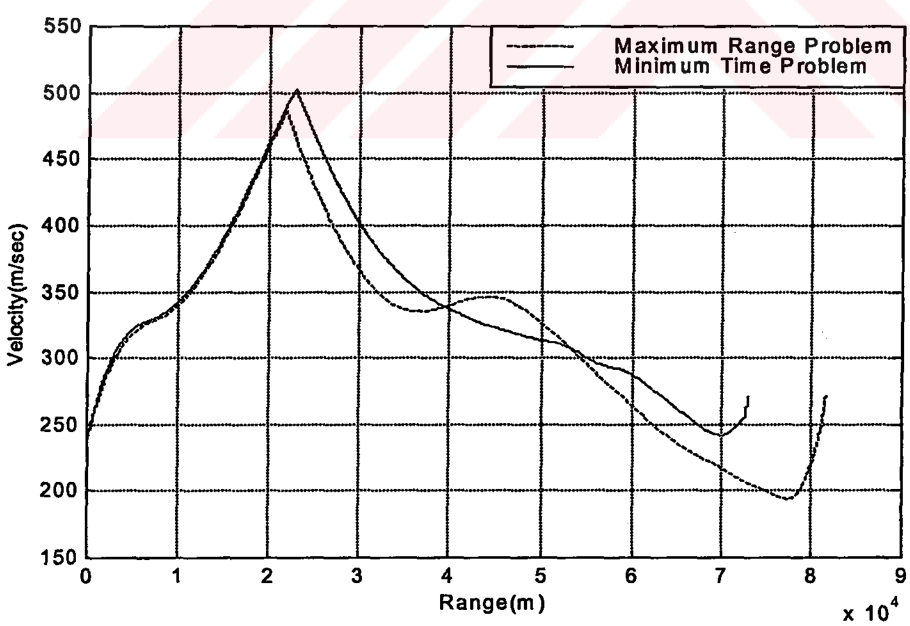


Figure B.10 Velocity - Range Graph for the Maximum Range, and Minimum Flight Time Trajectories

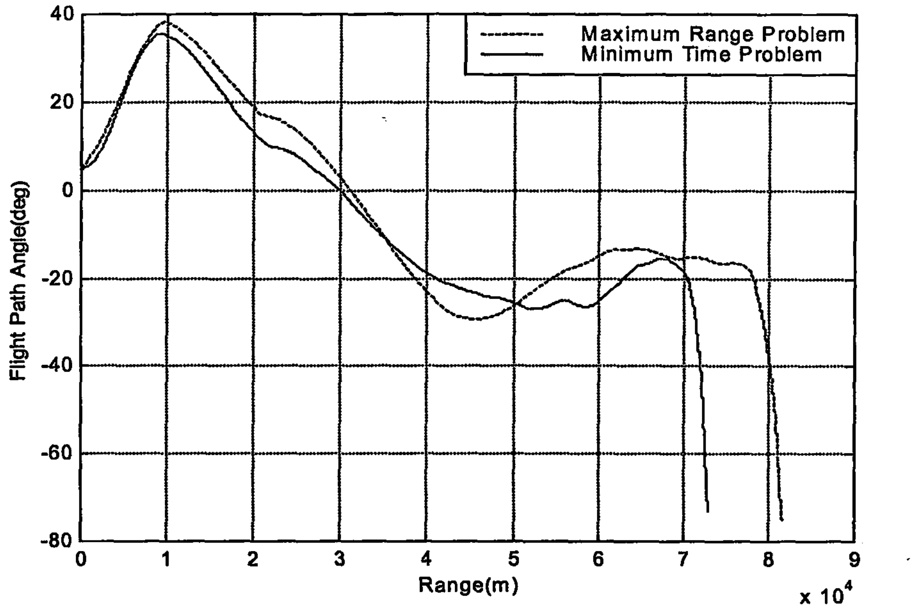


Figure B.11 Flight Path Angle - Range Graph for the Maximum Range and Minimum Flight Time Trajectories

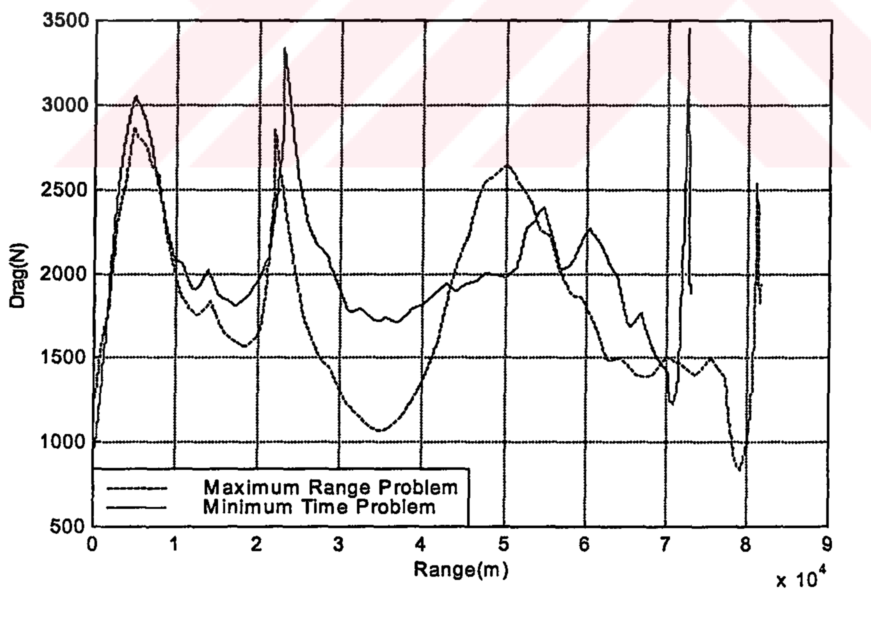


Figure B.12 Drag Force - Range Graph for Maximum Range and Minimum Flight Time Trajectories