Space trajectories optimization using variable-chromosome-length genetic algorithms

Ahmed H. Gad
Michigan Technological University

Follow this and additional works at: https://digitalcommons.mtu.edu/etds
Part of the Aerospace Engineering Commons

SPACE TRAJECTORIES OPTIMIZATION USING VARIABLE-CHROMOSOME-LENGTH GENETIC ALGORITHMS

By
Ahmed Hamdy Gad Elsayed Gad

A DISSERTATION
Submitted in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
Mechanical Engineering-Engineering Mechanics

MICHIGAN TECHNOLOGICAL UNIVERSITY
2011

This dissertation, “Space Trajectories Optimization Using Variable-Chromosome-Length Genetic Algorithms,” is hereby approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING – ENGINEERING MECHANICS.

Department of Mechanical Engineering – Engineering Mechanics

Signatures:

Dissertation Advisor: ______________________________
Ossama O. Abdelkhalik

Department Chair: ______________________________
William W. Predebon

Date: ______________________________

To my parents, my wife,
and
my sons Mohamed and Mostafa.

Contents

List of Figures

List of Tables

Acknowledgments

I wish to record my sincere appreciation and gratitude to my advisor Dr. Ossama Abdelkhalik for his continues support, encouragement, help, and advice. His valuable comments and objective criticizing were a continues source of challenge and made this dissertation a worthwhile endeavor.

I would like to express my deepest gratitude to my wife for her support. I wish also to thank my parents and family members whose support has enabled me to persevere through my life. May God reward them the highest of all rewards that are due them in their lives and their after.

Nomenclature

$\Delta\mathbf{v}$	=	impulsive maneuver velocity vector, km/s
Δv_T	=	total mission cost, km/s
T	=	time of flight, days
n	=	number of DSMs in a single leg
ε	=	epoch of a DSM as a fraction of transfer time
$\mathbf{r}$	=	heliocentric position vector in inertial frame, km
$\boldsymbol{v}$	=	heliocentric velocity vector in inertial frame, km/s
$\mathbf{v}_\infty$	=	hyperbolic velocity vector relative to the planet, km/s
v_∞	=	hyperbolic speed relative to the planet, km/s
$\boldsymbol{r}_{per}$	=	pericenter radius vector, km
r_{per}	=	pericenter radius, km
δ	=	deflection angle in the swing-by plane, rad
μ	=	gravitational constant, km^3/s^2
R	=	mean radius, km
η	=	swing-by plane rotation angle, rad
Π	=	perpendicular plane to the incoming relative velocity
Γ	=	intersection line between Π and the inertial ecliptic plane
Ω	=	angle between Γ and the inertial $\hat{I}$, rad
ι	=	inclination of Π to the ecliptic, rad
C	=	transformation matrix
m	=	number of swing-by maneuvers
P	=	planet identification number
F	=	fitness (cost function)
f	=	flight direction
t	=	Julian date

h	=	pericenter altitude, km
$\overline{h}$	=	normalized swing-by pericenter altitude
l	=	leg number
L_{max}	=	maximum chromosome length
i	=	maximum possible number of swing-by maneuvers
j	=	maximum possible number of total DSMs in the whole trajectory
k	=	maximum number of independent thrust impulses
z	=	number of swing-bys followed by a zero-DSM trajectory
q	=	number of bits
A	=	continuous design variable
B	=	discrete design variable
Δ	=	accuracy of continuous design variable
c	=	proportionality constant
L_s	=	chromosome length of sub-population
J_2	=	gravitational zonal harmonic
N_c	=	number of covered sites
M_r	=	ground track repetition period
e	=	eccentricity
i_n	=	inclination
ϑ_1	=	spacecraft's true anomaly above the first ground site
a	=	semi-major axis
N_r	=	number of successive orbit revolutions
n_m	=	orbital mean motion
$\Delta\phi$	=	changes in longitude
τ_E	=	Earth's sidereal rotational period
τ	=	spacecraft's nodal period
M	=	mean anomaly
ω	=	argument of perigee
ϕ	=	site longitude
λ	=	site latitude
ω_E	=	Earth spinning rate
ρ	=	angular radius of the Earth
λ_E	=	Earth central angle
η	=	nadir angle

ε_o	=	elevation angle
θ_{cov}	=	coverage angle
θ_{sh}	=	shadow angle
$\boldsymbol{r}_{sh}$	=	shadow position vector
T_{tsh}	=	target shadow period over a specific site
n_{sh}	=	number of successive shadow revolutions
N_m	=	number of Earth revolutions
$h_{perigee}$	=	perigee altitude
$\Delta\varphi_{shadow}$	=	change in shadow longitude
$\Delta\lambda_{shadow}$	=	change in shadow latitude

Subscripts

p	=	swing-by's planet
s/c	=	spacecraft
nps	=	non-powered swing-by
ps	=	powered swing-by
L	=	local frame
l	=	leg
req	=	required
min	=	minimum
max	=	maximum
s	=	sub-population
sh	=	shadow

Superscripts

$-$	=	incoming
$+$	=	outgoing

Abbreviations

CDV	=	Continuous Design Variables
DDV	=	Discrete Design Variables
DE	=	Differential Evolution
DSM	=	Deep Space Maneuver
DSMPGA	=	Dynamic-Size Multiple Population Genetic Algorithms
EM	=	Earth-Mars
EV	=	Earth-Venus
EVEJ	=	Earth-Venus-Earth-Jupiter
EVM	=	Earth-Venus-Mars
EVVEJS	=	Earth-Venus-Venus-Earth-Jupiter-Saturn
GA	=	Genetic Algorithms
GPOPS	=	General Pseudospectral OPtimal Control Software
GTOC	=	Global Trajectory Optimization Competition
GTOP	=	Global Trajectory Optimization Problems
HGGA	=	Hidden-Genes Genetic Algorithms
ITO-HGGA	=	Interplanetary Trajectory Optimization Using the Hidden Genes Genetic Algorithms
MESSENGER	=	Mercury Surface, Space Environment, Geochemistry, and Ranging
MGA	=	Multi Gravity Assist
MGADSM	=	Multi-Gravity-Assist Trajectory with Deep Space Maneuvers
MJD	=	Modified Julian Day
PSO	=	Particle Swarm optimization
RSTO	=	Repeated Shadow Track Orbits
TOF	=	Time of Flight

VCLGA	=	Variable Chromosome Length Genetic Algorithms
VE	=	Venus-Earth
VSDS	=	Variable-Size Design Space
VV	=	Venus-Venus

Abstract

The problem of optimal design of a multi-gravity-assist space trajectories, with free number of deep space maneuvers (MGADSM) poses multi-modal cost functions. In the general form of the problem, the number of design variables is solution dependent. To handle global optimization problems where the number of design variables varies from one solution to another, two novel genetic-based techniques are introduced: hidden genes genetic algorithm (HGGA) and dynamic-size multiple population genetic algorithm (DSMPGA).

In HGGA, a fixed length for the design variables is assigned for all solutions. Independent variables of each solution are divided into effective and ineffective (hidden) genes. Hidden genes are excluded in cost function evaluations. Full-length solutions undergo standard genetic operations. In DSMPGA, sub-populations of fixed size design spaces are randomly initialized. Standard genetic operations are carried out for a stage of generations. A new population is then created by reproduction from all members based on their relative fitness. The resulting sub-populations have different sizes from their initial sizes. The process repeats, leading to increasing the size of sub-populations of more fit solutions. Both techniques are applied to several MGADSM problems. They have the capability to determine the number of swing-bys, the planets to swing by, launch and arrival dates, and the number of deep space maneuvers as well as their locations, magnitudes, and directions in an optimal sense. The results show that solutions obtained using the developed tools match known solutions for complex case studies. The HGGA is also used to obtain the asteroids sequence and the mission structure in the global trajectory optimization competition (GTOC) problem.

As an application of GA optimization to Earth orbits, the problem of visiting a set of ground sites within a constrained time frame is solved. The J_2 perturbation and zonal coverage are considered to design repeated Sun-synchronous orbits. Finally, a new set of orbits, the repeated shadow track orbits (RSTO), is introduced. The orbit parameters are optimized such that the shadow of a spacecraft on the Earth visits the same locations periodically every desired number of days.

Chapter 1

Introduction

In recent years, many investigations have dealt with global optimization for space trajectory problems. Many spacecraft trajectory design problems can be formalized as optimization problems. Several different techniques have been proposed and tested on a variety of space trajectory optimization problems. Recently, Genetic Algorithm (GA) has been used to solve several orbital mechanics problems. Genetic algorithms are optimization techniques, based on the Darwinian principle of the survival of the fittest, which perform a stochastic search of initial conditions that maximize a given fitness function (1). The GAs have global search capability, and they do not require derivatives (2). GAs are particulary efficient for the type of problems where it is not necessarily to find the optimal solution, but rather a few reasonably good solutions.

In GA, a design point is called a chromosome (string). Each string consists of a set of independent design variables which define a candidate solution to the problem. Chromosomes are composed of genes (features), which take on different alleles (values) (1). Associated with each gene is its location in the chromosome. The initial population consists of a finite number of randomly generated individuals. GA performs a series of probabilistic operations on the current population to generate a new generation. Basic genetic operations are coding, evaluation, selection, crossover, mutation, and reproduction (1). At each generation, the fittest individuals (parents) are selected based on the fitness function. These individuals are then used to create the new generation (children). More fit populations are generated as a result of this process. Genetic algorithms have been used widely in the literature to solve several orbital mechanics problems. GAs have been used to solve the fuel-optimal spacecraft rendezvous problem (3), to design orbits with lower average revisit time over a particular target site (4), to design natural orbits for ground surveillance (5), to design constellations for zonal coverage (2), and to investigate the design of near-optimal low-thrust orbit transfers (6).

In this dissertation, two categories of space trajectories are investigated. The first is the interplanetary space trajectories. The problem of multi-gravity-assist space trajectory, with free number of deep space maneuvers (MGADSM) is formulated. Two novel optimization techniques, based on GA, are presented to handle global optimization problems where the number of design variables varies from one solution to another. The first technique is the hidden genes genetic algorithms, while the second is the dynamic-size multiple population genetic algorithm. Both techniques are used to investigate the complex problem of multi-variable-length genetic algorithm. The developed tools have the capability to find optimal interplanetary trajectories without a priori knowledge. As an application of the hidden genes concept, the problem of low-thrust propulsion missions is studied. The asteroids revisiting mission is investigated by exploring an optimal impulsive trajectory used as a preliminary scenario for the continuous thrust mission. The second category presented in this dissertation is the Earth orbiting trajectories. Two problems had been studied in this category. The first one is the problem of optimal orbit design to cover a given set of ground sites, within a constraint time frame. In the second problem, a new set of orbits, the repeated shadow track orbits (RSTO), is introduced.

1.1 Interplanetary Space Trajectories*

The optimization of interplanetary trajectories continues to receive a great deal of interest (7–10). In interplanetary missions, it is usually desired to send a spacecraft to rendezvous with a planet or an asteroid. The interplanetary trajectory design problem can be addressed either in a two-body or a three-body dynamics framework. It can be addressed assuming impulsive or continuous thrust. The patched conic approach for interplanetary trajectory design assumes a two-body dynamics model and the use of chemical propulsion system only (11). In patched conic mission design, it is observed that a deep space maneuver (DSM) may reduce the cost of a simple two-impulse interplanetary transfer (e.g. the cost of

* The material contained in this section has been accepted for publication in the *Journal of Spacecraft and Rockets* (7).

the Earth-Mars mission can be reduced by adding an impulse, almost mid-way, in addition to the initial and final impulses.) For a transfer between two non-coplanar orbits, a DSM reduces the out-of-plane component of the required impulsive velocity, and thus the total mission cost Δv_T (12). Due to $\mathbf{\Delta v}$ leveraging effect at large distances from the Sun, DSMs are used to reduce the required Δv_T, and hence, the equivalent propellant mass which allows for a larger size payload or a smaller launch vehicle. Navagh applied the primer vector theory to determine where and when to use a single DSM to reduce the cost of a trajectory (12). He applied DSMs to a Mars round trip mission to determine their effect on the launch opportunities. He also studied cycler trajectories and Mars mission abort scenarios. Later, Abilleira investigated the broken-plane maneuver impact on the total mission cost Δv_T, the incoming relative velocity, and the launch energy, for Earth-to-Mars trajectories (13). He suggested to apply the extra $\mathbf{\Delta v}$ close to the halfway point of the near 180 deg transfer. He mentioned that DSMs could also allow new families of trajectories that would satisfy very specific mission requirements not achievable with ballistic trajectories (13).

The optimization problem, in its general form, aims to minimize some cost function; usually the overall cost of the mission in terms of the fuel expenditure. Another possibility is to optimize mission trajectory to achieve minimum mission duration. Interplanetary missions usually employ gravity assist maneuvers to reduce the overall cost of the mission. The launch and arrival dates also affect the mission cost. The space trajectory optimization process must therefore determine the optimal values for the following parameters: the departure and arrival dates, the number of swing-bys, the planets to swing by, the number of deep space maneuvers and their magnitudes, directions, and locations. This optimization problem, in its general form, is challenging. Several optimization algorithms were developed in the literature. Izzo et al developed a deterministic search space pruning algorithm to investigate the problem of multiple gravity-assist (MGA) interplanetary trajectories design (14). The developed tool requires the user to specify the gravity assist sequence (i.e. the number of swing-bys and the planets to swing-by), and the pruning technique locates efficiently all the interesting parts in the search space of the DSMs variables (14). Olympio and Marmorat (15) studied the global optimization of multi gravity assist trajectories with deep space maneuvers (MGADSM). They implemented a pruning strategy on the search space to find fit trajectories. To that end, a stochastic initialization procedure, combined with a local optimization tool, were used to provide a set of locally optimal solutions. The primer vector theory was extended to study the multi gravity assist trajectories. The optimal number of DSMs was determined as well as their magnitudes and directions. This technique was verified using several interplanetary mission case studies. An efficient local optimization algorithm was applied to find a solution for complex problems (15). In this method, the user specifies the sequence of swing-bys a priori. Later, Olympio and Izzo applied the interaction prediction principle to decompose the MGA problem into sub-problems by introducing and relaxing boundary conditions (16). Parallel sub-problems could then be solved. The algorithm was able to efficiently calculate the optimum number of DSM impulses and their locations.

Evolutionary algorithms were implemented to solve the MGADSM problem (17, 18). Vasile and Pascale used an evolutionary algorithm with a systematic branching strategy to optimize the problem of MGADSM (17). They developed a design tool (IMAGO) that allowed for more exploration of the solution domain through balancing the local convergence and the global search. The search space was reduced by performing a deterministic step at every new run. The tool was applied several times on each specific problem to provide reliable results (17). IMAGO was able to calculate many typical optimal sequences for a Jupiter mission. For the complex trajectories, Cassini and Rosseta (17), IMAGO was used assuming a fixed planet sequence and wide ranges of design variables based on a priori knowledge of the solution space. Olds et al developed a trajectory optimization tool (MDTOP) to perform preliminary design of high-thrust interplanetary missions using the differential evolution (DE) strategy (18). They tuned the algorithm parameters to improve the DE's performance, and were able to solve complex interplanetary missions, such as Cassini and Galileo. A formulation for the N-impulse orbit transfer that integrates evolutionary algorithms and a Lambert's problem formalism results in a more efficient search algorithm (17, 19). This integration of Lambert's problem significantly reduces the size of the design space to include only candidate solutions which satisfy Lambert's problem solution. In that context, GA was implemented to obtain the optimum impulses for non-coplanar elliptical interplanetary orbit transfers (19, 20).

As can be seen from the previous discussion, often the mission designer determines the number of swing-bys and selects the planets to swing by by hand. The motivation of this study is to develop an optimization algorithm that can, without a priori knowledge, compute the number of swing-bys and the planets to swing by, in addition to the rest of the classical MGADSM design variables. This MGADSM optimization problem, in its general form, is challenging. On one hand, the objective function has multi minima which means a global optimization method is necessary for optimization. On the other hand, the objective function has a Variable-Size Design Space (VSDS). In a VSDS, the number of optimization variables varies depending on the specific solution. For instance, the number of variables in a candidate solution that has two swing-bys and one DSM is different from the number of variables of another candidate solution where the number of swing-bys is three and the number of DSMs is two. In addition to these challenges, the design variables in the MGADSM problem are mixed (some are continuous and some are discrete).

There are many types of evolutionary algorithms, such as particle swarm (PSO) (21) and differential evolution (DE) (17,18), which have been used to solve trajectory optimization problems. But neither of those methods handles discrete design variables. The fact that some of the design variables are discrete suggests the use of genetic algorithms (22, 23). Solution-dependent design variables mean that different solutions have different number of design variables. This fact hinders the implementation of standard genetic algorithms in optimization. Dasgupta and McGregor developed a structured genetic algorithm (sGA) (24). They introduced the terms of active and passive genes through applying a gene activation mechanism which utilizes a multi-layered structure for the chromosome. Kim and de Weck proposed a variable chromosome length genetic algorithm (VCLGA) in structural applications (25). The VCL concept depends on gradually increasing the chromosome length in subsequent stages to achieve progressive refinement. Increasing the chromosome length was proposed either by adding new design variables or by refining the current ones (25). Ryoo and Hajela developed a GA for structural topology optimization that handles variable chromosome lengths (26). Crossover between different lengths chromosomes is considered in their work. Brie and Morignot introduced a genetic planning system using variable chromosome length (27). The developed system is used to determine the optimal sequence of actions for a planing problem. Several genetic schemes are developed to improve the system performance such as complex fitness function, multipopulation, population reset, weak memetism, tournament selection and elitist genetic operators. The developed system has the ability to minimize the required memory and to handle unlimited plan size. Ahn and Ramakrishna developed a VCLGA to handle the shortest path routing problem (28). They introduced simple crossover and mutation operations work on variable chromosome length. The developed operations improved the solution quality and the coverage performance. Katari and others proposed an improved genetic algorithm to study data clustering problems (29). The improved GA used a modified crossover and mutation technique to improve the simulation coverage. Moreover, a variable length improved GA is implemented for image clustering which automatically finds the optimal numbers of cluster.

In this dissertation, the concept of hidden genes is introduced in genetic algorithm optimization (7). This concept allows the handling of all solutions in the design space as if they all have the same string length, and hence enables the implementation of standard genetic operations. This research presents an optimization algorithm and a software tool that have the capabilities to find, in an optimal sense, the values for the following design variables: optimal number of swing-bys, the planets to swing by, the times of swing-bys, the optimal number of DSMs, the components of these DSMs, the times at which these DSMs are applied, the optimal launch and arrival dates, and the optimal flight direction for the mission. The search space of the new algorithm includes solutions with muti-revolution trajectories. The software tool for interplanetary trajectory optimization using the hidden genes genetic algorithms (ITO-HGGA) is developed. Another technique is developed to investigate the VSDS problem. A novel dynamic-size multiple population genetic algorithm (DSMPGA) is implemented to handle the VSDS MGADSM Trajectory Optimization problem (10). Sub-populations, each has members of same size design space, are constructed. A genetic algorithm is implemented for a number of stages; each stage is a number of consequent generations in which the size of the sub-population remains fixed. At the end of each stage, the sizes of all sub-populations are changed based on the fitness of the members in each individual sub-population. This algorithm leads to the increase in the size of the sub-populations of more fit members. The DSMPGA can determine, without a priori knowledge, the number of swing-by and DSM maneuvers, the planets to swing-by, and the times of swing-bys and DSMs, in addition to the rest of the design variables.

As an application of the HGGA concept, the problem of GTOC5 is investigated. A global trajectory optimization technique is developed to solve the low-thrust propulsion mission. The optimization problem is considered for trajectories of a spacecraft rendezvous mission to a group of asteroids with revisiting. The problem of low-thrust trajectory optimization is usually formulated to minimize the propellent consumption. Hence, the greatest number of asteroids could be visited within the pre-specified mission duration. The problem is solved in four stages. At the first stage, an impulsive trajectory problem is solved to determine a preliminary mission scenario. The hidden genes genetic algorithm (HGGA) tool is used as an optimization technique (7). The second stage is then used to provide an initial guess for the continuous thrust trajectories. Standard genetic algorithm is used to calculate a poor continuous thrust trajectory for each leg. An optimum control module is then utilized in the third stage to calculate accurate continuous thrust trajectories. General pseudospectral optimal control software (GPOPS) is used in this stage. The obtained solution from the GPOPS module is a visible continuous thrust trajectory but with a limited number of nodes. This solution is then used as an initial guess in the final stage to compute the final detailed trajectory. In the forth stage, a constrained nonlinear optimization technique is used to provide the optimal trajectory data at one-day increments for each leg of the trajectory.

In this dissertation, Chapter 2 describes the MGADSM problem formulation. The ITO-HGGA tool is presented in Chapter 3. ITO-HGGA tool is tested for a number of MGADSM design problems and compared to known solutions in the literature. The complex problem of Cassini 2 mission is solved by the ITO-HGGA tool and presented in Chapter 3. Chapter 4 presents the implementation of the DSMPGA to the MGADSM problem. Numerical results are presented including the complex mission of Messenger (easy version). A solution for GTOC5 problem using the HGGA concept is presented in Chapter 5.

1.2 Earth Orbiting Trajectories*

In Newtonian mechanics, the natural motion of a spacecraft around the Earth is described by a second order vectorial differential equation, assuming that the spacecraft is attracted only by the Earth, and assuming that the Earth is a perfect sphere (11). The solution to these differential equations is either a circular, elliptic, parabolic, or a hyperbolic trajectory (31). The type of the trajectory is determined depending on how we initially place the spacecraft in orbit. The Earth is always at the focus of this conic trajectory. Regardless of the type of the trajectory, it is always possible to describe the orbit of the spacecraft using five parameters (32). Another parameter is needed to determine the position of the spacecraft on the orbit. A fundamental task in the design process of any space mission is to design the orbit(s) of the spacecraft. Designing an orbit means, then, finding the values for the five orbital elements such that the mission objectives are best achieved (33).

A wide range of applications require that a spacecraft passes over a given number of ground sites (23). Examples for this type of missions include: remote sensing (34), disaster monitoring (35), urban planning (36), natural resources, and ground surveillance missions

* The material contained in this chapter was previously published in the journal *Acta Astronautica, Elsevier* (23) and in the *International Journal of Aerospace Engineering* (30).

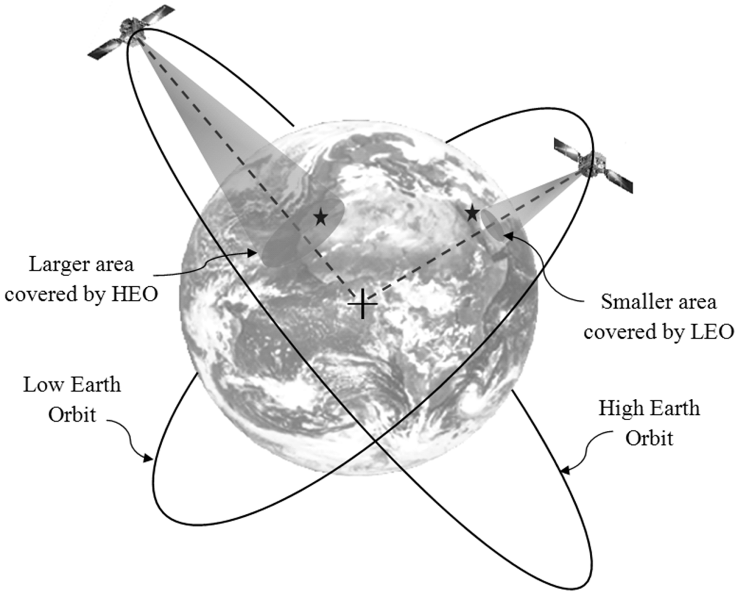

Figure 1.1: Spacecraft at high altitudes have larger coverage on Earth as compared to those at lower altitudes

(37). In this type of applications, the spacecraft is usually equipped with sensor(s) to take measurements for the ground sites of interest. The spacecraft does not have to visit (pass over) each site exactly; but rather a ground site is considered "visited" if the Field of View (FOV) of the sensor covers that ground site, at some point in time (38). As can be seen from Figure 1.1, the orbit selection dictates the coverage area on Earth surface. For spacecraft at high altitudes, the coverage area is bigger than that of a spacecraft in a low altitude orbit. On the other hand, the resolution of measurements from a low altitude spacecraft is better than the resolution that can be obtained from a higher altitude orbit, using the same sensor. The spacecraft orbit also determines the frequency of coverage, which is how often a given ground site will be visited. Besides the coverage area and the ground resolution, there are many other parameters that are affected by the orbit selection, e.g. the size of the spacecraft and the launch cost.

From the preceding discussion, it is clear that there are conflicting objectives that control the selection of the orbit. In many missions, mission designers use propulsion systems, mounted onboard the spacecraft, to maneuver the spacecraft continuously between ground sites. They specify the regions of interest and keep the spacecraft in a low altitude *nominal* orbit. The spacecraft is then controlled to maneuver between the regions of interest using the thrusters. This way, it becomes possible to collect high resolution measurements, and still visit all the ground sites, within a given time frame. There is an important research problem associated with this way of designing the space mission: what is the control strategy such that the fuel expenditure for maneuvering the spacecraft, between ground sites, is minimal. Several studies have addressed this problem. A representative study is the work developed by Guelman and Kogan (39), where they developed an optimal control algorithm to sort the sites in an optimal sense, and then maneuver the spacecraft between them. The results of their study show that the thrusters have to be working continuously throughout the mission lifetime. A typical example that they discussed is the case of 20 ground sites. The results show that a time frame of 50 days is needed to complete visiting all the sites (39). These solutions have major weaknesses: (1) they require thrusters to work continuously throughout the mission life time. This puts severe demands on the spacecraft systems, in terms of the power and fuel needed to run the thrusters, and hence dramatically increases the mission cost, and (2) this solution requires longer mission durations, due to the very low thrust level of the electric propulsion systems. For the case of 20 ground sites, the time needed to visit all of the sites is about 50 days using propulsion. If a natural orbit exists, all the sites will be visited in a much smaller time frame. These major weaknesses are mainly because these algorithms use propulsion systems, in some sense, to oppose the natural gravitational forces.

In previous works, a method to design natural orbits, to visit the regions of interest without the use of propulsion, was developed (5). A spacecraft, in a natural orbit, needs to be as close as possible to Earth when visiting a ground site to achieve high resolution. Yet, a synchronization is needed, between Earth rotational motion (spinning) and the spacecraft motion, to guarantee visiting all the ground sites. To find these natural orbits, two approaches were developed. In the first approach, the problem is formulated as an optimization problem. Stochastic optimization methods were used as an attractive alternative for optimizing space orbits design (8, 40). The cost function for this type of problems usually have numerous local minima. For this specific problem, a genetic algorithm technique was implemented for optimization (41). The second approach adopted a semi-analytical method to reduce the number of unknowns and then perform numerical search or stochastic optimization (42). The two methods suffer from the following weaknesses: (1) The two methods assume a two-body model for the spacecraft motion (A two-body model is a model that assumes the spacecraft is attracted only by the Earth, which is assumed a perfect sphere. In reality, this is not true. A more accurate model for the Earth gravitational field includes the effect of the Earth oblateness. This effect is modeled as zonal harmonic coefficients known as $J2$, $J3$, etc,) (2) both methods assume zero field of view for the spacecraft sensor. This assumption limits the number of solutions to the problem, and (3) the semi-analytical method assumes the existence of a solution. If a perfect solution does not exist, the method cannot return any solution even if there exists a solution with a small error. In recent developments, Kim et al investigated the possibility of using a genetic algorithm for finding the temporary target orbit, in order to reduce the average revisit time of an existing mission orbit, over a particular target site during a given time window (43). Genetic algorithms are also used to optimize the fuel consumption of Low Earth Orbit constellations for temporary reconnaissance missions (44), and also to minimize telecommunications coverage blackouts (45).

In this dissertation, Chapter 6 addresses the problem of optimal orbit design taking into consideration the second zonal harmonic of the Earth gravitational field, and taking into consideration the field of view of the sensor on board the spacecraft (23). Taking the gravitational perturbations into consideration in the optimization process allows for the design of repeated ground track sun-synchronous orbits. This type of orbits makes the spacecraft visits the same Earth site every given number of days, at the same day-time. On the other hand, taking the sensor's FOV into consideration in the optimization process results in more feasible solutions. A standard optimization tool that carries out coverage calculations is usually computationally expensive. The algorithm presented in Chapter 6 reduces the computational cost, while carrying out an optimization process, in two aspects: (1) the development in this research benefits from analytical developments to reduce the number of independent variables. A genetic algorithm is then developed to select the independent variables, in an optimal sense. The reduced number of independent variables leads to faster convergence and a reduced number of objective function evaluations, and (2) a fast numerical algorithm is developed to roughly compute the coverage state of a ground site. It is rough in the sense that it assumes that $J2$ is the only perturbing effect from the two-body model. Several objective functions can be used. In this research, an objective function that maximizes the number of visited ground sites and minimizes the time frame needed for the mission is presented.

Extremely large space structures are proposed for solar sail and space-based solar power generation missions (30). Solar sails, the size of San Francisco, are proposed to collect momentum from the solar radiation, for deep space missions. Solar Power Satellites (SPSs) are also large space structures orbiting the Earth. SPSs are proposed to generate electricity in space and transmit it to receivers on the Earth using either microwave or laser transmissions. For the purpose of cooling the Earth, and thereby reducing the global warming, it is also suggested to put a huge number of satellites at the first Lagrangian point between the Earth and the Sun, to provide shade for the Earth.

A great numbers of studies were conducted by NASA and the Department of Energy (DOE) during the seventies of the last century on the feasibility of the SPSs concept. The Office of Technology Assessment evaluated these studies and did not recommend immediate action toward implementation due to the technical challenges and high cost of the proposed mission at that time. In mid nineties, NASA took a fresh look at the concept; However they did not recommend, even discourage, further investments in this direction. Yet, an active interest toward implementing this concept can be seen nationally (46) and internationally (47, 48). Aiming at reducing the cost of the generated electricity from SPSs, in this research it is suggested to have multiple objectives for the same SPSs. SPSs orbiting Earth can provide shadow on ground for the Earth cooling purposes, generate electricity using its huge solar arrays, and save energy devoted to air conditioning in the shaded areas. Nations from the warm regions may participate in the cost of a SPS to get its shadow, if it can be provided to them on a regular basis. More than 70% of the operational expenses, for many buildings and facilities in warm regions, is dedicated for air conditioning. It has been observed that the main reason for these extremely high operational expenses is the direct solar rays from the Sun. The SPSs will serve as a Space-SunSetter for these regions participating in SPSs missions only. A Space-SunSetter will significantly reduce the operational expenses of every building in the shaded area.

There are two concepts for Space-SunSetter under investigation. The first is to design a huge spacecraft orbiting the Earth. The main challenge in this concept is to find an appropriate orbit satisfying a repeated shadow track over a specific region on Earth's surface. The second concept is a stationary Space-SunSetter. This concept depends on the idea of Space Elevator (49, 50). The purpose of the research in Chapter 7 is to investigate the feasibility of the first concept. Since this type of mission will require the Space-SunSetter shadow on ground to revisit the same place repeatedly, the orbital elements are developed to have a shadow visits a certain location on the Earth's surface for a given number of days, this is called the repeated shadow track condition (30). To find the orbit, among all repeated shadow track orbits, that has maximum duration time over a certain point on the Earth, an optimization tool was developed using GA.

Chapter 2

MGADSM Problem Formulation*

The problem of multi-gravity-assist space trajectory, with free number of deep space maneuvers (MGADSM) is formulated in this chapter (7, 10). The objective is to design an interplanetary trajectory for a spacecraft to travel from the departure planet to the target planet with a minimum cost. The spacecraft will benefit from as many as needed swing-bys of other planets. The spacecraft can also apply DSMs. The segment between any two planets is called a leg. A leg can have any number of DSMs, as seen in Figure 2.1. The spacecraft may have multiple revolutions about the Sun before proceeding to the next planet. The scenario of the mission refers to the sequence of swing-bys. The determination of the mission scenario then means the determination of the number of swing-bys, the planets of swing-bys, and the times of swing-bys. The problem is worked out in a two-body dynamics framework, and is formulated as an optimization problem.

The problem is formulated as follows: For given ranges for departure and arrival dates from the departure planet to a target planet, find the optimal selections for the number of swing-bys, the planets to swing by, the times of swing-bys, the number of DSMs, the components and directions of DSMs, the times at which DSMs are applied, and the exact launch and arrival dates, such that the total mission cost Δv_T is minimized. An essential step in any genetic optimization algorithm is to evaluate the cost function at different design points. At each design point, the optimization algorithm selects values for all the independent design variables of the problem. There are several other dependent variables that need to be computed before the cost function can be evaluated. This chapter details how these dependent

* The material contained in this chapter has been accepted for publication in the *Journal of Spacecraft and Rockets* (7).

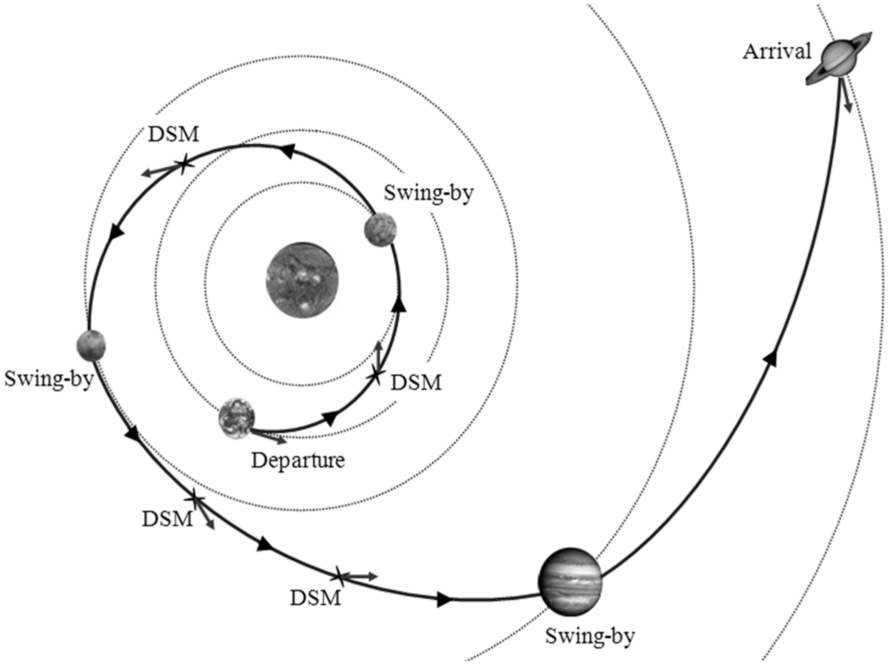

Figure 2.1: Multi-gravity assist trajectory with free number of deep space maneuvers (MGADSM).

variables are computed, given a set of values for the independent design variables.

2.1 N-impulses Trajectory

In the simple case of a two-impulse interplanetary orbit transfer, the total number of design variables is two (20). In this case, the trajectory has one leg between the departure and arrival planets. The two variables are the departure and arrival dates. For a candidate solution, the departure and arrival dates fix the time of flight T. A Lambert's problem is then solved to find the transfer orbit. The required departure and arrival impulses, $\Delta\mathbf{v}_d$ and $\Delta\mathbf{v}_a$, are then calculated.

In an N-impulses trajectory (no swing-bys), there are n deep space maneuvers in addition to the departure and arrival impulses. The number of different orbits is $n+1$. The number of unknowns in this case is $4(n+2)$. These unknowns are the impulse velocity increment vectors $\Delta\mathbf{v}$ and the times of applying them (t), for all impulses. The time of application of each DSM is defined as a fraction, ε, of the overall transfer time, T. So, $t_{DSM} = \varepsilon T$, where $0 < \varepsilon < 1$. Figure 2.2 shows a three-impulse trajectory (one DSM).

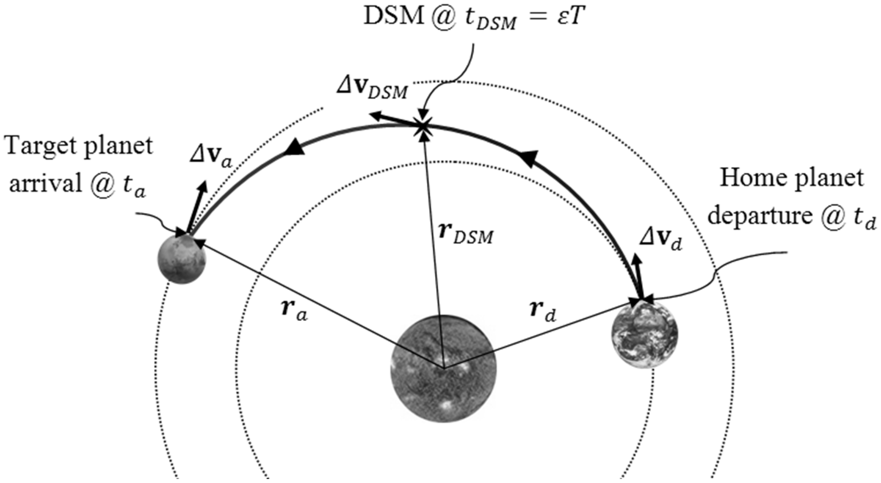
DSM @ $t_{DSM} = \varepsilon T$
$\Delta \mathbf{v}_{DSM}$
Target planet arrival @ t_a
$\Delta \mathbf{v}_a$
$\boldsymbol{r}_{DSM}$
Home planet departure @ t_d
$\Delta \mathbf{v}_d$
$\boldsymbol{r}_a$
$\boldsymbol{r}_d$

Figure 2.2: Three-impulse transfer orbit (single deep space maneuver).

Of the $4(n+2)$ unknowns, there are $4n+2$ independent design variables. These independent variables are selected to be: the departure and arrival dates, the velocity increment vector(s) at the first n impulses, and the fractional variable(s) ε at each DSM. The remaining 6 unknowns are computed as functions of the independent design variables as follows: the departure and arrival dates fix the departure and target planets positions, and hence the spacecraft heliocentric positions at these two locations, $\mathbf{r}_d$ and $\mathbf{r}_a$, are fixed. The impulsive velocity vector at departure, $\Delta\mathbf{v}_d$, yields the spacecraft initial velocity on the first transfer orbit, which along with $\mathbf{r}_d$ fixes the first transfer orbit. Once we have the first transfer orbit, and using the time of applying the first DSM, ε_1, the location of the first DSM is computed using Kepler's equation (51). The velocity of the spacecraft at that location before applying the DSM is also computed. The velocity increment at the first DSM is used to compute the spacecraft velocity right after the first DSM is applied. The procedure used in the first transfer orbit is repeated for all subsequent transfer orbits, but the last one. On the last transfer orbit, Lambert's problem is solved. The spacecraft positions at the last DSM and at arrival are known. The time of flight is also known. A Lambert solution yields the last transfer orbit. The velocity increments at the last DSM and at arrival are then computed. The total cost of the mission in this case is:

$$\Delta v_T = \|\Delta\mathbf{v}_d\| + \sum_{1}^{n} \|\Delta\mathbf{v}_{DSM}\| + \|\Delta\mathbf{v}_a\| \tag{2.1}$$

2.2 Gravity Assist Maneuvers

In this analysis, a gravity assist maneuver is assumed to be an instantaneous impulse applied to the spacecraft. The spacecraft heliocentric position vector is assumed not to change during the swing-by maneuver, and is equal to the swing-by planet heliocentric position vector at the swing-by instance.

$$\mathbf{r}^{-} = \mathbf{r}^{+} = \mathbf{r}_p \tag{2.2}$$

where $\mathbf{r}^{-}$ and $\mathbf{r}^{+}$ are the spacecraft incoming and outgoing heliocentric position vectors, respectively, and $\mathbf{r}_p$ is the swing-by planet heliocentric position vector.

As shown in Figure 2.3, the trajectory of the spacecraft relative to the planet is a hyperbola with the relative hyperbolic velocity vector $\mathbf{v}_\infty$ which is defined as (52):

$$\mathbf{v}_\infty = \mathbf{v}_{s/c} - \mathbf{v}_p \tag{2.3}$$

where $\mathbf{v}_{s/c}$ and $\mathbf{v}_p$ are the spacecraft and planet heliocentric velocity vectors at the swing-by instance, respectively.

Two types of swing-bys are implemented in this work: powered swing-bys and non-powered swing-bys (15). In a non-powered swing-by, the incoming and outgoing relative velocities, $\mathbf{v}_\infty^-$ and $\mathbf{v}_\infty^+$, respectively, have the same magnitude.

$$\|\mathbf{v}_\infty^-\| = \|\mathbf{v}_\infty^+\| = v_\infty \tag{2.4}$$

The swing-by plane is defined by the incoming relative velocity vector $\mathbf{v}_\infty^-$ and the pericenter radius vector $\mathbf{r}_{per}$. The change in the relative velocity direction in the swing-by plane, δ, can be computed from Equation (2.5) (52).

$$\sin(\delta/2) = \frac{\mu_p}{\mu_p + r_{per} v_\infty^2} \tag{2.5}$$

where μ_p is the gravitational constant of the swing-by planet, and $r_{per} = \|\mathbf{r}_{per}\|$. In a non-powered swing-by, the spacecraft velocity change is given by Equation (2.6) (15).

$$\|\Delta\mathbf{v}_{nps}\| = \|\mathbf{v}_\infty^+ - \mathbf{v}_\infty^-\| = 2v_\infty \sin(\delta/2) \tag{2.6}$$

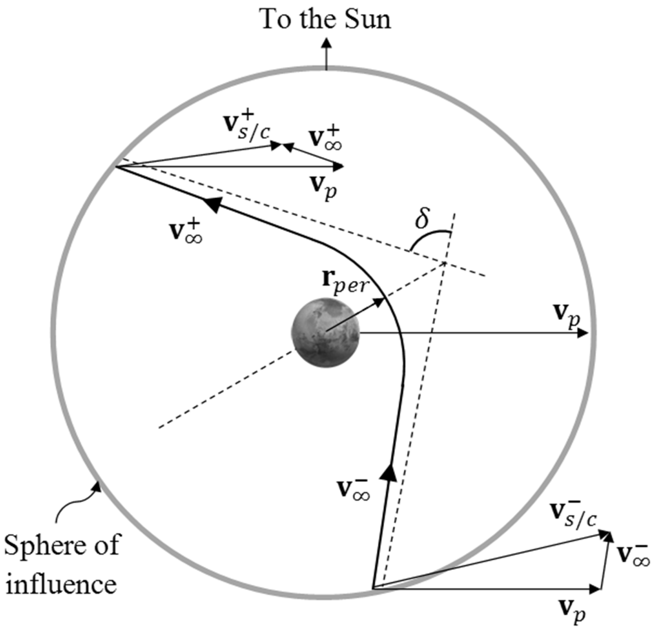

Figure 2.3: Gravity assist model as seen in the swing-by plane projection.

For a non-powered swing-by to be feasible, the periapsis radius must be higher than a minimum radius, i.e. $r_{per} > r_{per_{min}}$. In this work, we assumed $r_{per_{min}} = 1.1R_p$. This value is suitable for most solar planets, as Earth and Venus. A much higher periapsis radius is used in Jupiter swing-by to avoid radiation effects. If the desired gravity assist is not feasible via a non-powered swing-by, then an impulsive post-swing-by maneuver is applied (the powered swing-by). By applying a small impulse during the swing-by, higher deflection angles could be attained (16). The powered swing-by impulse can be computed using Equation (2.7).

$$\Delta\mathbf{v}_{ps} = (\mathbf{v}^{+}_{s/c})_{req} - (\mathbf{v}^{+}_{s/c})_{nps} \tag{2.7}$$

where $(\mathbf{v}^{+}_{s/c})_{req}$ is the required spacecraft heliocentric outgoing velocity vector; and $(\mathbf{v}^{+}_{s/c})_{nps}$ $= \mathbf{v}_p - \mathbf{v}^{+}_{\infty}$.

The swing-by plane needs to be computed in order to calculate $\mathbf{v}^{+}_{\infty}$. To define the swing-by plane, a rotation angle η is introduced (17). The vector $\mathbf{v}^{+}_{\infty}$ is obtained form the vector $\mathbf{v}^{-}_{\infty}$ by performing two consecutive rotations (δ and η). The incoming relative velocity vector $\mathbf{v}^{-}_{\infty}$ is defined in the heliocentric inertial frame $\hat{I}\hat{J}\hat{K}$. A local frame $\hat{i}\hat{j}\hat{k}$ is defined such that the unit vector $\hat{i}$ is in the direction of the incoming relative velocity vector; so

$$\hat{i} = \frac{\mathbf{v}_\infty^-}{\|\mathbf{v}_\infty^-\|} \tag{2.8}$$

The plane of the swing-by maneuver is the ij plane. Therefore, the outgoing relative velocity vector, as expressed in the local frame, $(\mathbf{v}_\infty^+)_L$, is:

$$(\mathbf{v}_\infty^+)_L = v_\infty[\cos\delta \quad \sin\delta \quad 0]^T \tag{2.9}$$

The $\mathbf{v}_\infty^+$ is computed in the inertial frame, via a coordinate transformation from the local frame to the inertial frame, as shown in Figure 2.4. To perform this transformation, the following procedures are applied. The perpendicular plane Π defined by its normal $\hat{i}$ intersects with the inertial ecliptic plane (IJ plane) in the line Γ. The direction of Γ depends on the orientation of the incoming relative velocity in the inertial frame. The angle between Γ and the inertial $\hat{I}$ is Ω which is defined in the IJ plane. The angle between Γ and the unit vector $\hat{j}$ is the rotation angle η which is defined in the perpendicular plane Π. The inclination of plane Π to the ecliptic plane is ι. A three-angle rotation is performed to calculate the local unit vector $\hat{j}$ in the inertial frame according to the following relation:

$$\hat{j} = \begin{bmatrix} \cos(-\Omega) & \sin(-\Omega) & 0 \\ -\sin(-\Omega) & \cos(-\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\iota) & \sin(-\iota) \\ 0 & -\sin(-\iota) & \cos(-\iota) \end{bmatrix} \begin{bmatrix} \cos(-\eta) & \sin(-\eta) & 0 \\ -\sin(-\eta) & \cos(-\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.10}$$

Finally, the outgoing velocity vector in the inertial frame is computed as follows:

$$\mathbf{v}_\infty^+ = C(\mathbf{v}_\infty^+)_L \tag{2.11}$$

where C is the transformation matrix:

$$C = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \end{bmatrix} \tag{2.12}$$

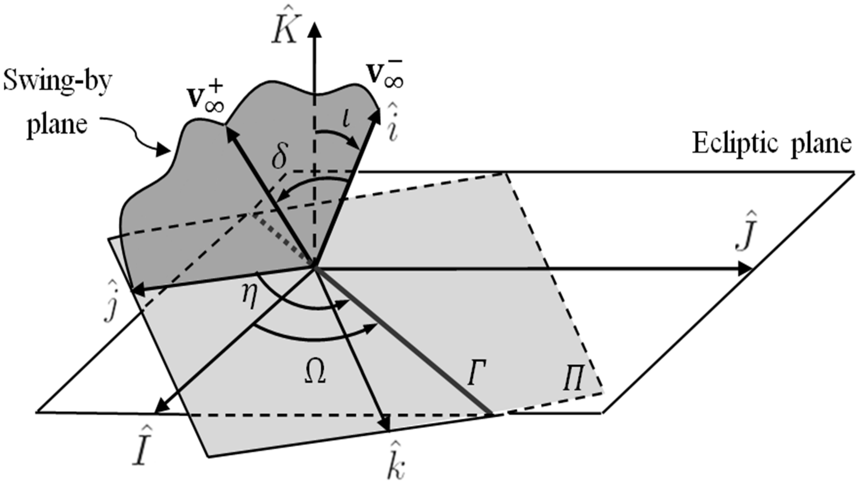

Figure 2.4: Transformation scheme from the local frame ijk to the inertial frame IJK showing the definition of the rotation angle η.

2.3 N-Impulses Multi Gravity Assist Trajectory

In this section, a formulation is introduced for the full problem of optimal trajectory design, including multiple gravity assist maneuvers, and possibly N-impulses in each leg. The mission is designed to transfer a spacecraft from a departure planet P_d to a target (arrival) planet P_a. Time windows are given for each of the departure and arrival dates. Consider a mission that consists of m gravity assist maneuvers. There are $m+1$ different legs in the trajectory. Each leg contains n_l deep space maneuvers. The time of flight for each leg, except the last leg, is an independent design variable. The calculations of the dependent variables are carried out starting from the departure planet, and from one leg to the next and so on.

In any trajectory leg, the spacecraft trajectory is solved as discussed in Section 2.1. The velocity vector at the leg end point is the heliocentric incoming velocity vector of the consequent gravity assist maneuver. If a leg has at least one DSM then the swing-by maneuver at the beginning of that leg is assumed a non powered swing-by maneuver. The swing-by independent design variables are the pericenter altitude h and the rotation angle η. The swing-by maneuver calculations are carried out, as discussed in Section 2.2, to yield the outgoing spacecraft velocity vector. This velocity vector is the spacecraft heliocentric velocity vector of the initial point on the consequent transfer trajectory. The process is repeated for all legs and swing-by maneuvers.

If any of the swing-by maneuvers is followed by a leg with no DSM, then that swing-by is assumed to be a powered swing-by maneuver. In this case, all dependent variables associated with the leg before the swing-by is calculated first. Then the dependent variables associated with the leg after the powered swing-by are computed. Finally the powered swing-by variables, including the swing-by impulse, are computed such that the two legs calculations are compatible. Assume the powered swing-by planet is planet P_l. The leg before planet P_l is leg l, and the leg after it is leg $l+1$. The times of swinging by planets P_l and P_{l+1} are known, and hence the positions of the spacecraft at these two times are known. There is no DSM in leg $l+1$, hence a Lambert's problem is solved to calculate the spacecraft velocity vectors on the initial and final points of leg $l+1$. The velocity vector at the initial point of leg $l+1$ is the required heliocentric outgoing velocity vector $(\mathbf{v}_{s/c}^{+})_{req}$ of the powered swing-by at planet P_l. To achieve this velocity, a swing-by impulse $\Delta\mathbf{v}_{ps}$ is added. Since it is desired to achieve all maneuvers with minimum fuel. We assume that the powered swing-by maneuver plane is the plane containing the incoming relative velocity vector $\mathbf{v}_{\infty}^{-}$ and the required outgoing relative velocity vector $(\mathbf{v}_{\infty}^{+})_{req}$, so that $\Delta\mathbf{v}_{ps}$ does not have an out of plane component.

$$(\mathbf{v}_{\infty}^{+})_{req} = (\mathbf{v}_{s/c}^{+})_{req} - \mathbf{v}_{p} \tag{2.13}$$

Equations (2.6, 2.7, and 2.13) are used to calculate $\Delta\mathbf{v}_{ps}$. The required deflection angle δ_{req} between $\mathbf{v}_{\infty}^{-}$ and $(\mathbf{v}_{\infty}^{+})_{req}$ is computed as:

$$\delta_{req} = \cos^{-1}\left(\frac{\mathbf{v}_{\infty}^{-}.(\mathbf{v}_{\infty}^{+})_{req}}{\|\mathbf{v}_{\infty}^{-}\|\|(\mathbf{v}_{\infty}^{+})_{req}\|}\right) \tag{2.14}$$

The maximum deflection angle δ_{max} is calculated from Equation (2.5) by substituting r_{per} with the permissable minimum periapsis radius $r_{per_{min}}$. If $\delta_{req} \leq \delta_{max}$, then $\mathbf{v}_{\infty}^{+}$ is in the same direction as $(\mathbf{v}_{\infty}^{+})_{req}$ and is calculated as:

$$\mathbf{v}_{\infty}^{+} = \mathbf{v}_{\infty}\frac{(\mathbf{v}_{\infty}^{+})_{req}}{\|(\mathbf{v}_{\infty}^{+})_{req}\|} \tag{2.15}$$

If $\delta_{req} > \delta_{max}$, then δ_{max} is used, and the swing-by maneuver is carried out in the same plane defined by $\mathbf{v}_{\infty}^{-}$ and $(\mathbf{v}_{\infty}^{+})_{req}$.

2.4 Summary

The problem of multi-gravity-assist space trajectory, with free number of deep space maneuvers (MGADSM) is introduced and formulated. The chapter addresses the DSM model and the MGA model which are used in the MGADSM formulation. The dynamic model of the MGADSM problem is presented. Two-body framework model is considered. The MGADSM model will be used later on to formulate the optimization problem. This chapter explains the dependent and independent design variables which are used to evaluate the objective (cost) function at different design points.

Chapter 3

Hidden Genes Genetic Algorithm for MGADSM Trajectories Optimization*

The problem of optimal design of a multi-gravity-assist space trajectory, with free number of deep space maneuvers, poses a multi-modal cost function (7). In the general form of the problem, the number of design variables is solution dependent. This chapter presents a genetic-based method developed to handle global optimization problems where the number of design variables vary from one solution to another. A fixed length for the design variables is assigned for all solutions. Independent variables of each solution are divided into effective and ineffective segments. Ineffective segments (hidden genes) are excluded in cost function evaluations. Full-length solutions undergo standard genetic operations. This new method is applied to several interplanetary trajectory design problems. This method has the capability to determine the number of swing-bys, the planets to swing-by, launch and arrival dates, and the number of deep space maneuvers as well as their locations, magnitudes, and directions, in an optimal sense. The results presented in this chapter show that solutions obtained using this tool match known solutions for complex case studies.

* The material contained in this chapter has been accepted for publication in the *Journal of Spacecraft and Rockets* (7).

3.1 Hidden Genes Genetic Algorithm

The independent design variables to be optimized are described in Chapter 2. The objective is to minimize the total cost, Δv_T, of the trajectory for a MGADSM mission that consists of m gravity assist maneuvers and n_l deep space maneuvers in each leg of the $m+1$ mission's legs. Equation (3.1) shows the total cost of the mission.

$$\Delta v_T = \|\Delta \mathbf{v}_d\| + \sum_1^m \|\Delta \mathbf{v}_{ps}\| + \sum_1^{m+1} \sum_1^{n_l} \|\Delta \mathbf{v}_{DSM}\| + \|\Delta \mathbf{v}_a\| \tag{3.1}$$

where $\Delta \mathbf{v}_d$ and $\Delta \mathbf{v}_a$ are the departure and arrival impulses, respectively, $\Delta \mathbf{v}_{ps}$ is the post-swing-by impulse of the powered gravity assist only, and $\Delta \mathbf{v}_{DSM}$ is the applied deep space maneuver impulse. The fitness F at a design point, which is maximized to determine the fittest solution, is defined as:

$$F = \frac{1}{\Delta v_T} \tag{3.2}$$

GA is adopted for optimization to find the most fit solutions. Each individual in the GA population represents a set of independent design variables. Those variables are divided into two categories: discrete and continuous variables, as seen in Table 3.1. Each individual is presented in a binary format as a binary string. This string contains the binary representation of all the design variables values at the corresponding design point. The number of bits for each variable determines its accuracy. For the CDVs, the number of bits q_A for a continuous variable A is selected according to the following inequality:

$$2^{q_A} \leq \frac{A_{max} - A_{min}}{\Delta_A} + 1 \qquad ; \qquad A_{max} \neq A_{min} \tag{3.3}$$

where A_{min} and A_{max} are the lower and upper bounds of the variable A, and Δ_A is the desired accuracy of A. For the DDVs, each variable would be assigned a unique binary string. The number of bits q_B for a discrete variable B depends on the lower and upper bounds of the variable, B_{min} and B_{max}, respectively. The variable q_B is computed as:

$$\begin{aligned} & q_B = 1\,\text{bit} && \text{if} \quad B_{max} = B_{min} \\ \textit{or}, \qquad & 2^{q_B} \leq B_{max} - B_{min} + 1 && \text{if} \quad B_{max} \neq B_{min} \end{aligned} \tag{3.4}$$

Table 3.1
The discrete and continuous independent design variables of the MGADSM problem

Discrete Design Variables (DDV)	Continuous Design Variables (CDV)
No. of swing-by maneuvers, m	Departure date, t_d
Swing-by planets, P_1,..., P_i	Arrival date, t_a
The count of DSMs in each leg, n_1,..., n_{i+1}	Time of flight, T_1,..., T_i
Flight direction, f	Normalized pericenter altitudes, $\overline{h}_1$,..., $\overline{h}_i$
	Rotation angles, η_1,..., η_i
	Epochs of DSMs, ε_1,..., ε_j
	DSMs, $\Delta\mathbf{v}_1$,..., $\Delta\mathbf{v}_k$

In Table 3.1, i is the maximum possible number of swing-by maneuvers and j is the maximum possible number of total DSMs in the entire trajectory (both i and j are specified by the user). The term DSM is used to define any thrust impulse applied during the mission course except the launch, arrival, and powered swing-by impulses. The maximum number of independent thrust impulses is k, which can be explained as follows: if n_l is the maximum number of DSMs in a leg, then there are n_l independent thrust impulses if this leg is the first one; while for the consequent legs, the number of required independent thrust impulses is $n_l - 1$. Each thrust impulse $\Delta\mathbf{v}$ consists of three continuous design variables. The pericenter altitudes h are normalized with respect to the mean radius of the associated swing-by planet, R_p. The normalized pericenter altitudes $\overline{h}$ are used as design variables to limit the resulting pericenter altitudes to feasible values only, when dealing with different swing-by planets with obvious varied radii. The epoch of a DSM, ε, specifies the time at which the impulsive maneuver is applied, as a fraction of the associated leg transfer time. For most applications, a DSM could be applied from 10% up to 90% of the total time of flight in the associated leg. The flight direction f of the whole mission is either retrograde or posigrade.

Given that the maximum possible number of swing-by maneuvers is i, then we create i discrete design variables P_1, P_2, ..., P_i. Each variable P_l determines the planet about which the l^{th} swing-by occurs, as shown in Figure 3.1. The range of the discrete variable P_l is 1 through 8, which are the indices for the planets in the solar system, starting from Mercury to Neptune. The order of the swing-bys is the same as the order of the variables P_l, as shown in Figure 3.1. If the selected number of swing-bys is $m < i$, then the first m variables (P_1, P_2, ..., P_m) will be selected.

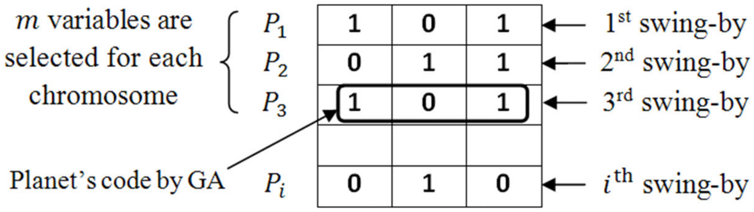
m variables are selected for each chromosome
P_1 1 0 1 ← 1st swing-by
P_2 0 1 1 ← 2nd swing-by
P_3 1 0 1 ← 3rd swing-by
Planet's code by GA
P_i 0 1 0 ← ith swing-by

Figure 3.1: Coding of swing-by planets as descrite design variables in HGGA.

For a MGADSM problem, the total number of independent design variables depends on the number of swing-by maneuvers, m, and the number of DSMs in every leg, $n_1, ..., n_{i+1}$. Selecting different values for the discrete design variables $m, n_1, ..., n_{i+1}$, changes the mission scenario and the number of DSMs in each leg. So, for a given solution, the values of these variables dictate the length of a certain portion in the chromosome of that solution. For example, if $m = 2$, then we need to allocate two swing-by portions in the chromosome; however, if $m = 3$, then three positions will be needed, for swing-bys, in the chromosome. Thus, changing the mission scenario and/or the number of DSMs is accompanied by a change in the length of the chromosome. Standard genetic algorithms cannot handle this problem because of the variation of strings lengths among different solutions. Genetic operations, such as crossover, are defined only for fixed-length string populations. To overcome this problem, the concept of hidden genes is introduced.

Let L_{max} be the length of the longest possible chromosome (this chromosome corresponds to a trajectory in which the spacecraft performs the maximum possible number of swing-bys and applies the maximum number of DSMs). All chromosomes in the population are allocated a fixed length equal to L_{max}, which is computed, based on the design variables stated in Table 3.1, as in Equation (3.5).

$$L_{max} = 5 + 3i + j + 2(i - z) + 3k \tag{3.5}$$

where z is the number of swing-by maneuvers which are followed by a no-impulse leg. From the upper bounds of the key design variables (m, n), the maximum chromosome length could be calculated a priori. In a general solution (a point in the design space), some of the variables in the chromosome will be ineffective in cost function evaluation; the genes describing these variables are here referred to as hidden genes. The hidden genes, however, will be used in the genetic operations in generating future generations. Figure 3.2 shows typical solution chromosomes where the variables at the top of the figure are the

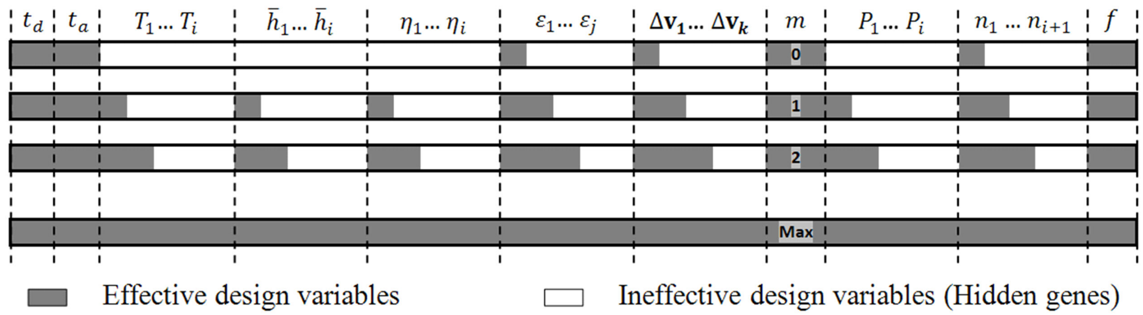

Figure 3.2: Typical chromosomes for the trajectory optimization problem in ITO-HGGA tool; each row represents a chromosome.

independent design variables. Each row in Figure 3.2 represents a member (single individual) in the GA population. The dark parts represent genes that are effective in fitness function evaluations. The white parts represent the hidden genes.

Because all chromosomes have the same length, standard definitions of genetic algorithms operations can still be applied to this problem. Hidden genes will take part in all genetic operations like normal genes. To illustrate this concept, an example on the crossover operation, between two parent solutions to generate two children (new) solutions, is here presented. Figure 3.3 shows two parent chromosomes and the resulting chromosomes after crossover, where a single point crossover is performed. Consider, for example, the genes representing the number of swing-by maneuvers and the genes representing the number of DSMs. The genes representing the number of swing-by maneuvers and the numbers of DSMs are always effective genes. The chromosome length allows for up to 5 swing-bys and up to 2 DSMs in each leg. In the first parent, the genes representing the first two swing-bys are effective genes, while there are hidden genes representing the other three swing-bys. Similarly, the genes representing the numbers of DSMs of legs 1 through 3 ($n_1 : n_3$) are effective genes, while there are hidden genes representing the rest of DSMs variables. In the second parent, the genes representing the first three swing-bys are effective genes, while the genes representing the other two swing-bys are hidden. The genes representing the first 4 DSM's variables are effective genes ($n_1 : n_4$), while there are hidden genes representing the rest of DSM variables. The crossover operation is carried out as defined in the standard genetic algorithm. The resulting two children are shown in Figure 3.3. The first child has effective genes representing a single swing-by and two DSM variables. The effective genes in the second child correspond to a zero-swing-by maneuver with a single DSM maneuver. The rest of the chromosomes of the children are hidden genes which represent the ineffec-

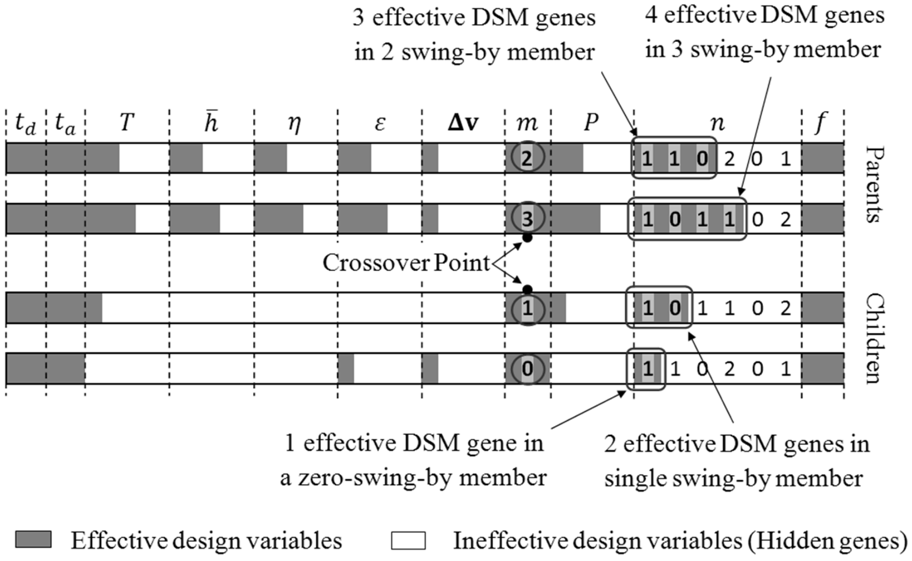
3 effective DSM genes in 2 swing-by member
4 effective DSM genes in 3 swing-by member
t_d
t_a
T
$\bar{h}$
η
ε
$\mathbf{\Delta v}$
m
P
n
f
2
1 1 0 2 0 1
3
1 0 1 1 0 2
Parents
Crossover Point
1
1 0 1 1 0 2
0
1 1 0 2 0 1
Children
1 effective DSM gene in a zero-swing-by member
2 effective DSM genes in single swing-by member
■ Effective design variables
□ Ineffective design variables (Hidden genes)

Figure 3.3: Crossover operation in hidden genes genetic algorithm.

tive design variables. Therefore, the generated children could have different scenarios and DSM sequences from those of the origin parents.

This algorithm is tested for several interplanetary missions ranging from simple to complex missions. The optimization algorithm (ITO-HGGA tool) is designed to find the mission scenario (the number of swing-bys and the planets to swing by) as well as the rest of the independent design variables: the times of swing-by, the number of DSMs, the times of DSMs, the magnitudes/directions of DSMs, and the departure/arrival dates. The size of the design space is controlled by the bounds of the independent design variables.

To reduce the computational cost for complex missions, the problem may be solved by two-phase approach. A trajectory design optimization can be started by assuming no DSMs in the trajectory (zero DSMs trajectory). This reduces the number of independent design variables by eliminating the following design variables: periapsis altitudes, rotation angles, epoch of DSMs, and thrust impulses. This reduction in the number of design variables allows for exploring wider ranges for each of the remaining design variables. Specifically, we can open the search space for any number of swing-bys with any planets in the solar system; wider ranges of departure and arrival dates and times of flight can be used; and both possible flight directions can be considered. This will result in a set of fit scenarios (zero DSM solutions). For each one of these scenarios, we then allow DSMs to be added to the trajectory and optimize our selection for these DSMs. So, we optimize on the number of DSMs in each leg and their locations/magnitudes/directions, while maintaining the scenario fixed. The departure/arrival dates and times of flight are allowed to change as design variables, with narrow ranges around the values obtained from the zero DSM solution. This technique of solving the problem by the two-phase approach has the advantage of reducing the computational time. This reduction in computational time comes at a price; most of the design space will not be explored. In general, optimizing a given scenario through adding DSMs, improves the fitness of this trajectory (If a zero DSM trajectory is optimal, then the second step in the optimization process will add no DSMs to the scenario). On the other hand, some solutions are fit only when there are DSMs in the trajectory and the fitness of the corresponding zero DSM scenario is poor (e.g. the MESSENGER mission trajectory is fit when we have a DSM in the first leg and becomes poor if we assume no DSM in the first leg). This means that it is not possible to find the zero DSM solution among the fittest solutions in the first step. For this kind of trajectories, it is not possible to find the optimal solution by solving the problem using the two-phase approach.

As an example, consider the Cassini 2 mission trajectory: let the maximum possible number of swing-by maneuvers be four, and let only one impulse, as a maximum, be applied in each leg. In optimizing all the design variables, the required number of independent design variables is 33 (11 DDVs and 22 CDVs). This is a computationally expensive problem, especially with wide ranges for the design variables. On the other hand, by solving a zero-DSM problem, the number of design variables is reduced to only 12 variables (6 DDVs and 6 CDVs). This step solves for the optimum mission's scenario without deep space maneuvers. Then, the second step is performed with 27 design variables (5 DDVs and 22 CDVs). The ranges of the CDVs in the second step is reduced based on the information from the initial mission's scenario.

The ITO-HGGA tool optimizes also the number of revolutions in each leg. The time of flight (TOF) of each leg and the time of applying each DSM are selected as design variables to allow for multi-revolution transfers. Lambert's problem is solved once in each leg. Based on the position vectors and the TOF, Lambert's solution may have both single and multi-revolution transfers. The selection criterion is as follows: the selected lambert's transfer should minimize the former maneuver cost. This maneuver could be a departure impulse, a DSM, or a swing-by maneuver. In the final leg, the selected lambert's transfer will minimize the former maneuver cost plus the arrival impulse cost.

A Matlab toolbox (GENETIC v2.1) is used (53). GENETIC v2.1 is structured such that continuous, discrete, and mixed continuous/discrete problems can be addressed. The main operations in the basic GA are coding, evaluation, selection, crossover and mutation. GA tuning parameters depend on the specific problem. A uniform crossover operation is conducted with a probability varied from 0.9 to 0.98. The mutation probability is selected between 0.01 and 0.08. Roulette wheel is used in selection operation. Proportional ranking is implemented in most cases. In general, linear ranking is implemented with higher crossover probability values, in solving a zero-DSM problem, in order to increase the diversity in the population. The linear ranking is implemented to avoid local minima traps. To maintain a diverse population, niching principle is applied by degrading the fitness of the similar individuals (54). A simple niching technique is implemented to solve the complex case of Cassini 2 mission. The solution obtained by the genetic algorithm is not necessary an optimal solution, nor at a local minimum. Therefore, a constrained nonlinear optimization technique is used to improve the solution by finding the closest local minimum to that solution. The local optimizer is only optimizing over the continuous design variables, not the discrete variables. The genetic algorithm solution is used as an initial guess in the local search algorithm.

3.2 Numerical Results

This section presents a number of case studies for interplanetary space missions trajectory optimization. Comparisons to other solutions in the literature is presented to validate the obtained results.

3.2.1 Earth-Mars Mission

A MGADSM trajectory is optimized for the Earth-Mars mission (EM). The lower and upper bounds, for all design variables, are listed in Table 3.2. The maximum possible number of swing-by maneuvers is selected to be two. The maximum number of DSMs in each leg is selected to be one impulses. As can be seen from Table 3.2, a gravity assist maneuver could be performed with any planet in the solar system, from Mercury to Neptune. The direction of flight can be posigrade or retrograde. The time of flight of each leg, except the last one, is selected between 40 and 300 days.

Table 3.2
Bounds of design variables for Earth-Mars mission

Design Variables	Lower Bound	Upper Bound
No. of swing-by maneuvers, m	0	2
Swing-by planets identification numbers, P_1,..., P_i	1 (Mercury)	8 (Neptune)
No. of DSMs in each mission's leg, n_1,..., n_{i+1}	0	1
Flight direction, f	Posigrade	Retrograde
Departure date, t_d	01-Jun-2004	01-Jul-2004
Arrival date, t_a	01-Apr-2005	01-Jul-2005
Time of flight (days)/leg, T_1,..., T_i	40	300
Swing-by normalized pericenter altitude, $\overline{h}_1$,..., $\overline{h}_i$	0.1	10
Swing-by plane rotation angle (rad), η_1,..., η_i	0	2π
Epoch of DSM, ε_1,..., ε_j	0.1	0.9
DSM (km/s), Δv_1,..., Δv_k	-5	5

The total number of design variables for this mission is 21 (7 DDVs and 14 CDVs). Wide ranges for the design variables are adopted, as listed in Table 3.2. A population of 500 individuals is used, for 300 generations. A local optimizer uses the fittest GA solution as an initial guess to find a local minimum. The resulting solution has a single swing-by maneuver at Venus, with a total cost of 10.754 km/s. The fittest trajectory has a single DSM in the first leg as shown in Table 3.3.

The same problem is solved again by dividing the optimization process into two steps to reduce the required computational time. First, a zero-DSM solution is sought to determine a mission scenario. The number of design variables in this first step is only 8 variables (4 DDVs and 4 CDVs). ITO-HGGA, with 200 populations and 100 generations, is used. The convergence of the optimization algorithm is shown in Figure 3.4. The resulting zero-DSM scenario is a single swing-by maneuver at Venus with a cost of 10.788 km/s. A local optimization tool is then used to improve the solution to the nearest local minimum solution. The local optimizer reduces the cost to 10.783 km/s. A powered swing-by is implemented in this case and the required impulse of the swing-by is 0.002 km/s. Figure 3.5 shows the zero-DSM Earth-Venus-Mars trajectory.

In the second step, the problem is solved using the full MGADSM formulation, assuming that the mission scenario is EVM (the scenario obtained from the first step). The ranges for the departure, swing-by, and arrival dates are varied within only 10 days around those values obtained from the first step. The number of design variables in this step is 12 (2 DDVs and 10 CDVs). A population of 300 individuals has been used for 100 generations of GA followed by a local minima optimizer. The result of the second optimization step is

Table 3.3
MGADSM solution for the EVM mission (ITO-HGGA)

Mission Parameter	**MGADSM Scenario**
Departure date,t_d	05-Jun-2004 00:28:38
Departure impulse (km/s)	4.53
DSM date	11-Sep-2004 18:35:44
DSM impulse (km/s)	0.1293 (t_d + 98.77 days)
Venus Swing-by date	19-Nov-2004 01:42:40
Post-swing-by impulse (km/s)	2.27×10^{-12}
Pericenter altitude (km)	7937.913
Arrival date	15-May-2005 15:07:38
Arrival impulse (km/s)	6.095
Time of flight (days)	167.07 , 177.6
Mission duration (days)	344.67
Mission Cost (km/s)	**10.754**

Table 3.4
MGADSM solution trajectory for the EVM mission using the two-phase approach technique (ITO-HGGA tool)

Mission Parameter	**Zero-DSM Model**	**MGADSM Model**
	Initial Estimate	Final Scenario
Departure date,t_d	05-Jun-2004 01:52:21	02-Jun-2004 11:43:25
Departure impulse (km/s)	4.616	4.457
DSM date	-	22-Aug-2004 07:07:46
DSM impulse (km/s)	-	0.1801 (t_d + 80.81 days)
Venus Swing-by date	20-Nov-2004 15:10:59	19-Nov-2004 07:02:55
Post-swing-by impulse (km/s)	0.002	1.68×10^{-13}
Pericenter altitude (km)	7996.782	7869.048
Arrival date	14-May-2005 13:18:06	16-May-2005 03:29:08
Arrival impulse (km/s)	6.165	6.091
Time of flight (days)	168.56 , 174.92	169.81 , 177.85
Mission duration (days)	343.48	347.66
Mission Cost (km/s)	**10.783**	**10.728**

a single DSM of 180.1 m/s in the first leg at 80.81 days from mission start time, as shown in Figure 3.5. The total transfer cost is slightly reduced to 10.728 km/s in the second optimization step. Table 3.4 shows the zero-DSM trajectory as well as the final trajectory. As shown in Table 3.4, the effect of the DSM is reducing the departure and arrival impulses, as well as reducing the impulse of the powered swing-by to almost zero. The fittest trajectory could be considered as a non-powered gravity assist trajectory with a single DSM.

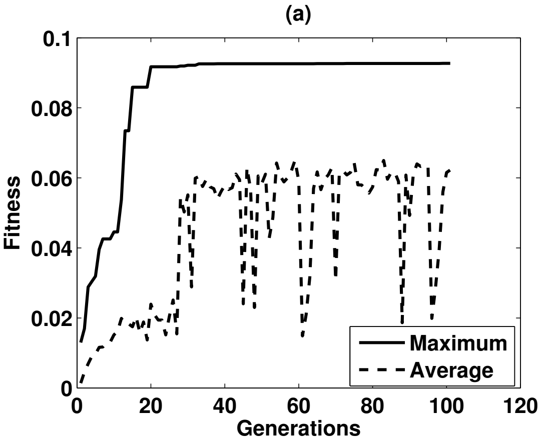

Figure 3.4: Convergence of the ITO-HGGA tool for the EVM mission: a) zero-DSM model, b) MGADSM model.

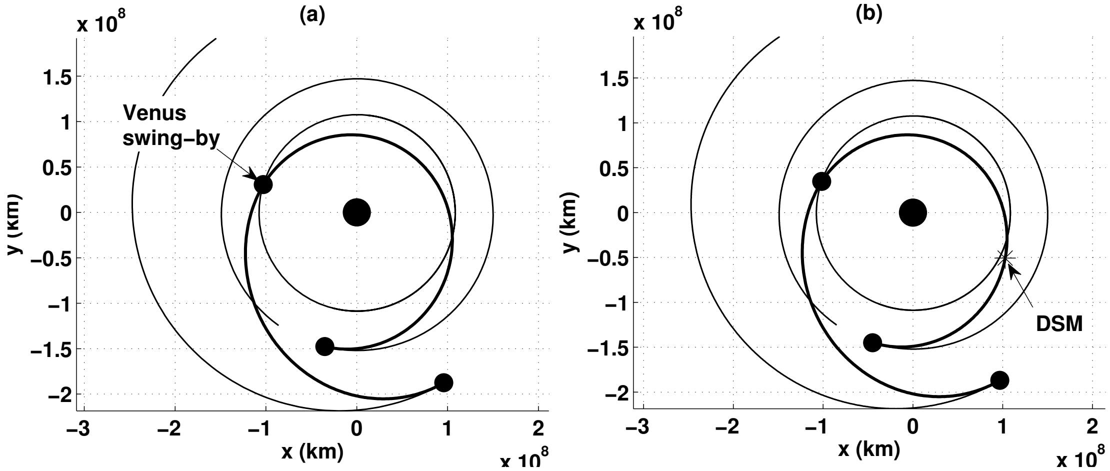

Figure 3.5: Optimal EVM mission using ITO-HGGA tool: a) zero-DSM model, b) MGADSM model.

3.2.2 Earth-Jupiter Mission

A mission to Jupiter is considered. It is desired to find the MGADSM trajectory with minimum cost. Wide ranges of all design variables are allowed in optimization to explore more of the design space. The design variables' bounds are listed in Table 3.5. The total number of design variables is 33 variables (7 DDVs and 26 CDVs). The zero-DSM model is used to determine an initial scenario. The independent design variables in this step are 8 variables (4 DDVs and 4 CDVs). A population of 300 individuals and 100 generations are used in GA optimization.

Table 3.5
Bounds of Earth-Jupiter mission's design variables

Design Variables	Lower Bound	Upper Bound
No. of swing-by maneuvers, m	0	2
Swing-by planets identification numbers, P_1,..., P_i	1 (Mercury)	8 (Neptune)
No. of DSMs in each mission's leg, n_1,..., n_{i+1}	0	2
Flight direction, f	Posigrade	Retrograde
Departure date, t_d	01-Sep-2016	30-Sep-2016
Arrival date, t_a	01-Sep-2021	31-Dec-2021
Time of flight (days)/leg, T_1,..., T_i	80	800
Swing-by normalized pericenter altitude, $\bar{h}_1$,..., $\bar{h}_i$	0.1	10
Swing-by plane rotation angle (rad), η_1,..., η_i	0	2π
Epoch of DSM, ε_1,..., ε_j	0.1	0.9
DSM (km/s), Δv_1,..., Δv_k	-5	5

Table 3.6
Optimal MGADSM trajectory of EVEJ mission using ITO-HGGA

Mission Parameter	Zero-DSM Model	MGADSM Model	
	Initial Estimate	Scenario (1)	Scenario (2)
Departure date	09-Sep-2016 11:38:03	06-Sep-2016 13:36:17	07-Sep-2016 01:55:17
Departure impulse (km/s)	3.653	3.542	3.439
DSM date	-	-	21-Feb-2017 08:29:43
DSM impulse (km/s)	-	-	0.109
Venus swing-by date	05-Sep-2017 05:57:07	05-Sep-2017 14:57:28	07-Sep-2017 07:43:57
Post-swing-by impulse(km/s)	0.0004	-	2.38e-014
Pericenter altitude (km)	1402.2	1307.28	613.545
DSM date	-	14-May-2018 09:31:08	-
DSM impulse (km/s)	-	0.0002	-
Earth swing-by date	30-Mar-2019 02:25:00	30-Mar-2019 03:14:06	29-Mar-2019 02:19:05
Post-swing-by impulse(km/s)	0.443	0.441	0.444
Pericenter altitude (km)	637.8	637.8	637.8
Arrival date	18-Sep-2021 21:15:27	24-Sep-2021 23:59:59	17-Sep-2021 07:43:51
Arrival impulse (km/s)	6.202	6.195	6.19
Time of flight (days)	360.76, 570.85, 903.79	364.05, 570.51, 909.87	365.24, 567.77, 903.23
Mission duration (days)	1835.4	1844.43	1836.24
Mission Cost (km/s)	**10.298**	**10.178**	**10.182**

The resulting fittest scenario is a two swing-by trajectory with swing-bys around Venus then Earth (EVEJ), as shown in Figure 3.6. The solution is a posigrade multi-revolution trajectory. The total cost of this zero-DSM trajectory is 10.298 km/s, as shown in Table 3.6. The scenario of this solution is then used in the second optimization step. The two swing-bys are fixed (EVEJ). The departure/arrival dates and the time of flight design variables obtained from the first step are allowed to vary within 10 days from their values obtained from the first step.

A population of 500 individuals and 100 generations is used. Figure 3.7 shows the convergence of the optimization algorithm. Table 3.6 shows the zero DSM solution, as

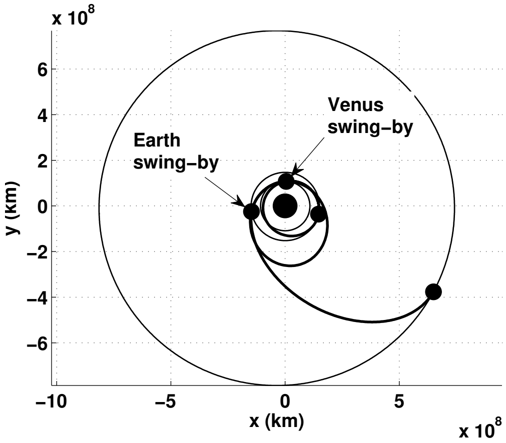

Figure 3.6: Optimal zero-DSM trajectory for Earth-Jupiter mission using ITO-HGGA tool.

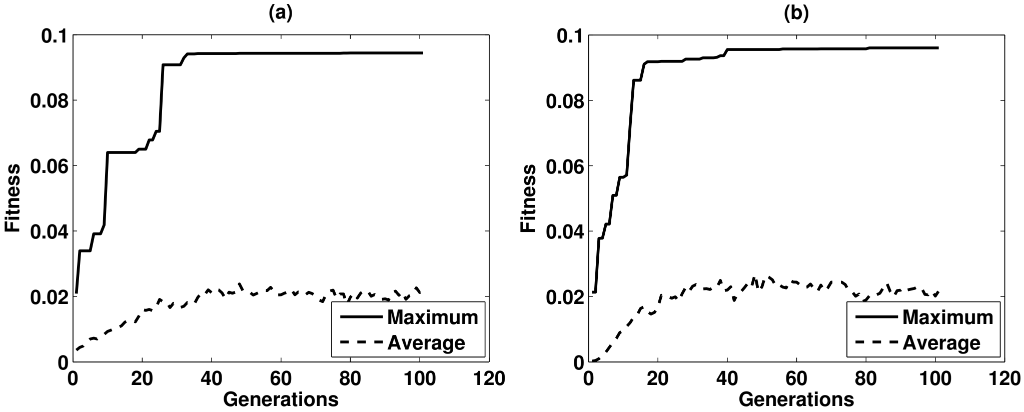

Figure 3.7: Convergence of the ITO-HGGA tool for Jupiter mission: a) zero-DSM, b) MGADSM.

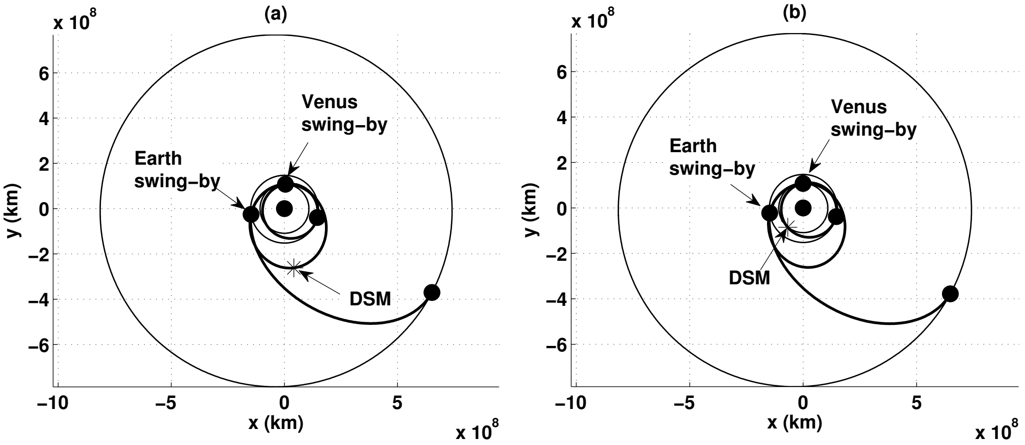

Figure 3.8: Optimal EVEJ mission using ITO-HGGA tool: a) case 1 (10.178 km/s), b) case 2 (10.182 km/s).

well as two solutions with DSMs. One solution has a single DSM in the second leg (VE), while the other solution has a single DSM in the first leg (EV), as shown in Figure 3.8. In the first scenario, the DSM amplitude is 0.2 m/s which seems an insignificant value with respect to the total trajectory cost. It is expected that this small DSM may be vanished with more iterations.

3.2.3 Earth-Saturn Mission (Cassini 2)

One of the most complicated multi gravity assist trajectories is the Cassini 2 trajectory. In 1997, The Cassini-Huygens mission was launched to study the planet Saturn and its moons (55). The ITO-HGGA tool is used to search for a minimum-cost trajectory for an Earth-Saturn trip. For the sake of making comparisons with the literature, a narrow range of departure date is allowed, around the known published date for the Cassini 2 mission. The ranges for the other design variables are wide enough to investigate all possible solutions. Table 3.7 presents the upper and lower bounds for the design variables. A zero-DSM model is initially used to find an initial mission scenario. There are 12 independent design variables in this step (6 DDVs and 6 CDVs). A population of 500 individuals and 1500 generations is used. Initial attempts show that resulting solutions are always trapped in regions that are worse than the known solutions for cassini 2 mission. This suggests allowing more time (more generations) for ITO-HGGA to search for more solutions. To avoid high computational cost, a simple niching technique is implemented (54). The GA convergence is shown in Figure 3.9.

The obtained mission scenario is a four swing-by trajectory with the same planet sequence as of the actual Cassini 2 mission scenario (EVVEJS). The trajectory is a posigrade transfer with 10.685 km/s total transfer cost, as shown in Table 3.8. Then, the initial zero-DSM scenario is used in the MGADSM model to obtain the final trajectory. The planet sequence is fixed at EVVEJS, with narrow ranges for the departure, arrival, and gravity assist dates.

A population of 500 individuals and 1000 generations is used. Then, a local optimizer is used. Twenty-seven design variables are used in this model (5 DDVs and 22 CDVs). The GA convergence is shown in Figure 3.9. The final solution is presented in Table 3.8. The trajectory has two deep space maneuvers, one in the first leg (EV) and the other is in the second leg (VV). The total transfer cost is reduced to 8.385 km/s after applying the deep

Table 3.7
Bounds of Cassini 2 mission's design variables

Design Variables	**Lower Bound**	**Upper Bound**
No. of swing-by maneuvers, m	1	4
Swing-by planets identification numbers, P_1,..., P_i	2 (Venus)	5 (Jupiter)
No. of DSMs in each mission's leg, n_1,..., n_{i+1}	0	1
Flight direction, f	Posigrade	Retrograde
Departure date, t_d	01-Nov-1997	31-Nov-1997
Arrival date, t_a	01-Jan-2007	30-Jun-2007
Time of flight (days)/leg, T_1,..., T_i	40	1000
Swing-by normalized pericenter altitude, $\bar{h}_1$,..., $\bar{h}_i$	0.1	10
Swing-by plane rotation angle (rad), η_1,..., η_i	0	2π
Epoch of DSM, ε_1,..., ε_j	0.1	0.9
DSM (km/s), Δv_1,..., Δv_k	-5	5

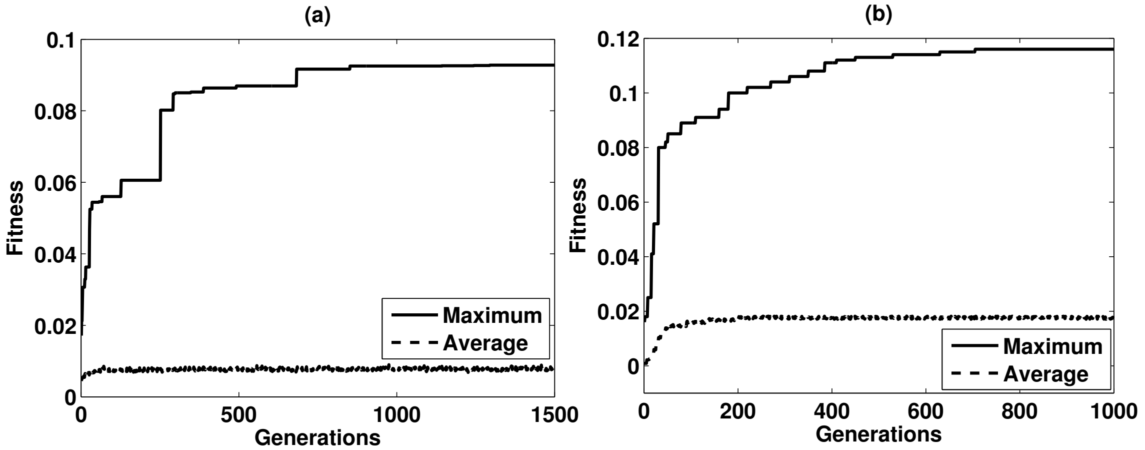

Figure 3.9: Convergence of the ITO-HGGA tool for Cassini 2 mission: a) zero-DSM model, b) MGADSM model.

space maneuvers. The zero-DSM initial trajectory and the MGADSM final trajectory are shown in Figure 3.10.

3.3 Comparisons and Discussion

The cost being optimized in this work is a function of discrete and continuous design variables. The discrete variables determine the mission scenario (swing-by planets), the number of DSMs in each leg, and the direction of flight (posigrade, retrograde). The continuous variables determine the dates and parameters of the events. The ITO-HGGA tool imple-

Table 3.8
Optimal MGADSM trajectory of Cassini 2 mission using ITO-HGGA

Mission Parameter	**Zero-DSM Model** Initial Estimate	**MGADSM Model** Final Scenario
Departure date	30-Nov-1997 00:00:00	13-Nov-1997 11:12:37
Departure impulse (km/s)	3.779	3.293
DSM date	-	25-Mar-1998 22:27:03
DSM impulse (km/s)	-	0.449
Venus Swing-by date	20-May-1998 12:58:21	30-Apr-1998 04:17:03
Post-swing-by impulse (km/s)	2.633	-
Pericenter altitude (km)	27470.963	2590.174
DSM date	-	11-Dec-1998 14:55:31
DSM impulse (km/s)	-	0.396
Venus Swing-by date	26-Jun-1999 14:04:56	27-Jun-1999 11:24:49
Post-swing-by impulse (km/s)	1.096×10^{-05}	2.21×10^{-06}
Pericenter altitude (km)	605.303	245.216
Earth Swing-by date	19-Aug-1999 09:05:31	19-Aug-1999 16:18:04
Post-swing-by impulse (km/s)	1.36×10^{-06}	6.04×10^{-08}
Pericenter altitude (km)	1810.704	1975.905
Jupiter Swing-by date	24-Mar-2001 00:21:22	31-Mar-2001 09:30:36
Post-swing-by impulse (km/s)	1.89×10^{-04}	1.68×10^{-07}
Pericenter altitude (km)	5167772.8	4918886.8
Arrival date	12-Jan-2007 12:50:27	23-Mar-2007 21:31:17
Arrival impulse (km/s)	4.273	4.247
Time of flight (days)	171.54, 402.05, 53.79, 582.64, 2120.52	168.67, 423.3, 53.2, 589.72, 2182.54
Mission duration (days)	3330.54	3417.43
Mission Cost (km/s)	**10.685**	**8.385**

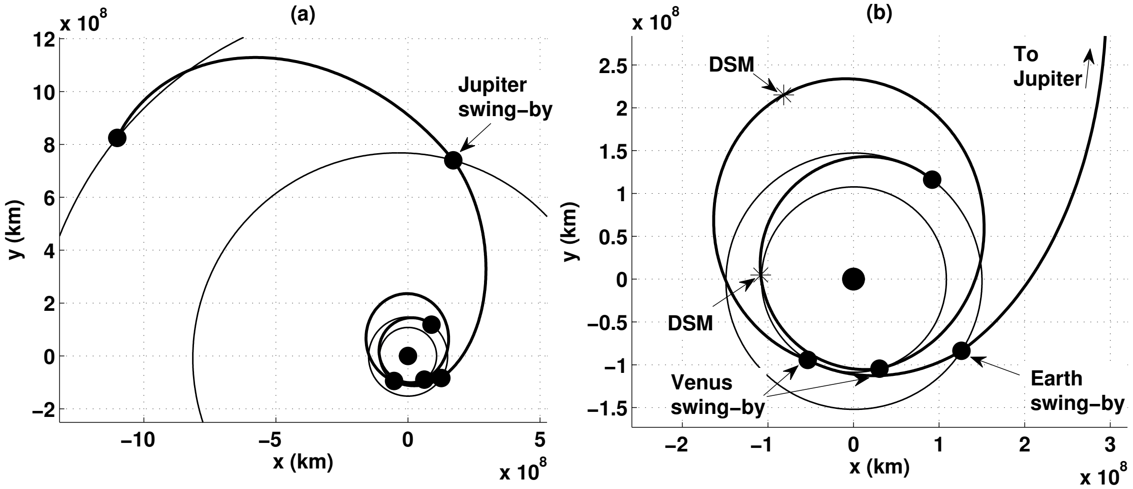

Figure 3.10: Optimal Cassini 2 mission using ITO-HGGA: a) zero-DSM model, b) MGADSM model (inner planets only).

mented in this dissertation finds the values for all the design variables in the best solution trajectory. Then the mission scenario, the direction of flight, and the number of DSMs are fixed, and the continuous variables are tuned by a local optimizer. The local optimizer implements a constrained nonlinear optimization technique. The proposed hidden genes concept is different from the VCL-GA concept presented in Reference (25). The VCL-GA assumes the same size for all chromosomes in a given population. The chromosome length, however, may vary from one generation to another. On other hand, the proposed hidden genes concept handles problems where chromosomes of different lengths may exist in the same population.

The Earth-Venus-Mars mission trajectory optimization has been addressed in the literature (15). The solution presented in Reference (15) was obtained using the extended primer vector theory, and has a single swing-by maneuver and a single DSM. In implementing the primer vector method, the departure and arrival dates were assumed fixed (the mission duration is 340 days). The Venus swing-by time was also constrained to occur at 165 days from departure. The resulting solution has a DSM of 68.7 m/s in the first leg at 96.08 days from mission start date. The total cost of the mission is 10.786 km/s (15). The solution obtained using the ITO-HGGA tool has also one swing by Venus and one DSM of 180.1 m/s in the first leg at 80.81 days from launch date. The total cost of the mission is 10.728 km/s, as shown in Table 3.4. The reduction in the total cost obtained using the ITO-HGGA tool, as compared to that of Reference (15), is accompanied by changes in the mission's launch, swing-by, and arrival dates, without significantly changing the total mission duration. This solution is obtained by allowing the maneuver's launch, swing-by, and arrival dates to be freely chosen during the zero-DSM optimization step.

Reference (15) presents a minimum-cost solution trajectory for the Earth-Jupiter mission, assuming the fixed planet sequence EVEJ. The departure, arrival, and swing-by dates were also assumed fixed, with a launch in 2016 and a mission duration of 1862 days. The primer vector theorem solution has four DSMs. Two DSMs are applied in the first two legs. The total transfer cost for this solution is 10.267 km/s. The ITO-HGGA tool presented in this chapter is able to find automatically the known swing-by sequence for the Jupiter mission in 2016. As listed in Table 3.6, the initial scenario (zero-DSM trajectory) is obtained, starting from wide ranges for the design variables. Then, the MGADSM solutions are obtained based on that initial scenario. Each of the two MGADSM solutions presented in Table 3.6, have less total cost than the solution presented in Reference (15). There is only one DSM in each of the two solutions. The solution presented in Reference (15) requires four DSMs.

The Cassini 2 mission trajectory design problem has been addressed in several studies (15–18), where it is always assumed that a fixed swing-by sequence (EVVEJS) is known. In Reference (17), the IMAGO obtained a solution that has a single DSM, and a total cost of 9.06 km/s. Reference (15) implements the space pruning technique and finds the values and locations of the DSMs to minimize the total transfer cost. The departure, arrival, and swing-by dates were fixed. The minimum cost obtained in (15) is 8.877 km/s. Olympio and Izzo (16) recently developed an algorithm to find the optimal DSM structure in a given trajectory scenario. For the Cassini 2 planets' sequence described in the GTOP database (55), Reference (16) found a trajectory with a total cost of 8.387 km/s. The solution obtained using the ITO-HGGA tool finds a solution with a total cost of 8.385 km/s (as seen in Table 3.8), which is very close to the reported result in the GTOP database (8.383 km/s) (55), and is also very close to the solution presented in Reference (16). The ITO-HGGA tool in this case, however, finds the planets sequence as well as the DSM structure.

Initially, the ITO-HGGA tool could not converge to the known planet sequence of the Cassini 2 mission in the first optimization phase. But rather, it converged to a worse solution. It has been observed that, in phase one, the solutions are trapped around a certain planet sequence that is worse than the known optimal scenario. The known optimal scenario (EVVEJS) could be obtained by forcing the last swing-by to be planet Jupiter. This suggested running the ITO-HGGA tool for more generations so that the optimal scenario is obtained. In order to avoid high computational cost and yet increase the diversity in the population, a niching algorithm is implemented (54). A niching technique applies a fitness degradation to the row fitness function such that fitness are depressed in the regions where solutions have already been found (54). In ITO-HGGA, a simple niching algorithm is added. Every five generations, the fitness are degraded for the current fittest solution, and all other solutions that have a similar planet sequence to the fittest solution. By implementing this niching technique, the ITO-HGGA tool is able to find the known optimal scenario in the first phase of optimization.

Figures 3.4, 3.7, and 3.9 show the convergence of the developed ITO-HGGA tool for the three case studies discussed in this chapter. As can be seen from the figures, the number of generations needed for convergence to the solution varies from one case to another. In the Earth-Venus-Mars mission, forty generations are needed in both the zero-DSM phase

(8 design variables) and in the MGADSM phase (12 variables). In the Cassini 2 mission, the number of design variables is 12 and 27 for the first and the second phases, respectively. The zero-DSM phase converges in 900 generations, while the MGADSM phase converges in 700 generations. Therefore, the much complex mission requires much higher number of populations and generations to converge to the optimal solution.

3.4 Summary

A novel methodology is proposed to tackle the optimization problems with variable size design space. A hidden gene genetic algorithm tool is developed to handle the optimization of MGADSM problems. This new tool has the capability to optimize the multiple gravity assist scenario, as well as the DSM structure. A zero-DSM model is initially solved to calculate an initial estimate scenario. The swing-by planets sequence is then fixed and narrow intervals of design variables are used to investigate the MGADSM model. The developed algorithm is verified by solving simple and complex interplanetary trajectories. The results show that the developed methodology is robust enough to tackle the complexity of the MGADSM problem. Moreover, the developed tool has the advantage of effectively determine accurate solutions in low computation cost.

Chapter 4

Dynamic-Size Multiple Population Genetic Algorithm for MGADSM Trajectories Optimization*

The problem of optimal design of a multi-gravity-assist space trajectory, with free number of deep space maneuvers, in its general form, poses a multi-modal objective function which design space size is variable (10). This chapter presents a genetic-based method developed to handle global, variable size design space, optimization problems where the number of design variables vary from one solution to another. Sub-populations of fixed size design spaces are randomly initialized. Standard genetic operations are carried out for a stage of generations. A new population is then created by reproduction from all members in all sub-populations based on their relative fitness. The resulting sub-populations have, in general, different sizes from their initial sizes. The process repeats, leading to increasing the size of sub-populations of more fit solutions and decreasing the size of sub-populations of less fit solutions. This method has the capability to determine the number of swing-bys, the planets to swing by, launch and arrival dates, and the number of deep space maneuvers as well as their locations, magnitudes, and directions, in an optimal sense. This new method is applied to several interplanetary trajectory design problems. The results presented in this chapter show that solutions obtained using this tool match known solutions for complex case studies.

* The material contained in this chapter has been accepted for publication in the *Journal of Guidance, Control, and Dynamics* (10).

4.1 Dynamic-Size Multiple Population Genetic Algorithm Optimization Technique

Genetic operations, such as crossover, are defined only for fixed-length string populations. Many engineering optimization problems, including the MGADSM problem, poses variable-length strings. To overcome this problem, the concept of Dynamic-Size Multiple Population (DSMP) is introduced. A software tool for interplanetary trajectory optimization using the Dynamic-Size multiple population genetic algorithms (DSMPGA) is developed in this study. The concept of the DSMPGA is to create an initial population that consists of sub-populations, as shown in Figure 4.1. All sub-populations are processed in parallel. All the members in each sub-population will have the same chromosome length. Different sub-populations will have different chromosome lengths. The standard genetic operations will be applied to each sub-population, for a certain number of generations (a stage of generations) as shown in Figure 4.1. The fitness of all members in the overall population are evaluated at the end of each stage. The selection operation is then carried out across the board (from all sub-population), based on members' fitness, regardless of the chromosome length. The more fit members from all sub-populations will be reproduced to subsequent stages, in their respective sub-populations. Hence, the size of each sub-population will change from one stage to another. Sub-populations that have higher (lower) fitness members will increase (decrease) in size. Sub-populations are then processed for another stage of generations, a selection at the end of the new stage is then carried out based on the members' fitness, and so on.

This approach will enable the use of standard genetic operations in VSDS problems. Figure 4.1 shows an illustration for this concept. Each symbol represents solutions of the same chromosome length. Sub-populations, each include the same symbol, are stacked in a one big population. Sub-populations will change their sizes from one stage to another based on the members's fitness, leading to the evolution of more fit members. There are two options for the sizes of the initial sub-populations (number of individuals in each sub-population.) The first option is to start with equal sizes for all sub-populations. The second option, which is adapted in this chapter for the MGADSM problem, is to start with an initial size for each sub-population that is proportional to the number of design variables in that sub-population. This choice is selected based on the fact that a problem with higher number of design variables need a bigger size population for better convergence. The

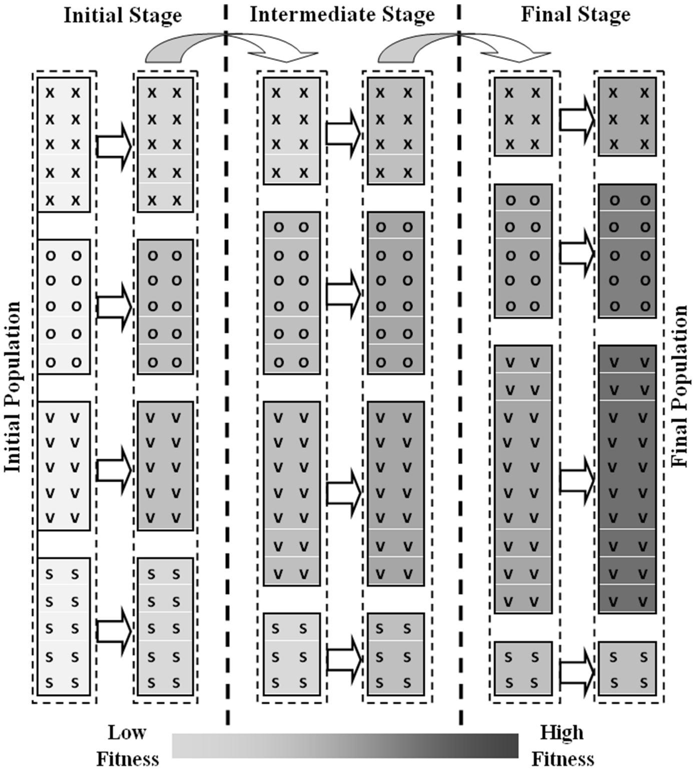

Figure 4.1: Illustration of the dynamic-size multiple population genetic algorithm concept

proportionality constant is *c*. The size of each sub-population then varies in sub-sequent generations. The overall population size, however, remain constant, and is equal to the summation of all initial sub-populations' sizes.

At the end of each stage, an evaluation process is performed to re-size all sub-populations based on the relative fitness of all individuals in all sub-populations. A roulette selection method is applied to choose parents for the next stage from the overall population of the current stage. The roulette method simulates a roulette wheel with the area of each segment proportional to its fitness. A higher fitness individual has a bigger segment area. A random number is then used to select one of the sections with a probability equal to its area. The selected individuals are separated according to their associated sub-population. To guarantee that the fittest individuals of each sub-population survive to the next stage, a small portion of the fittest individuals, elite count, is specified. The elite count is selected to be 10% in each sub-population. These elite individuals guarantee that each sub-population will be presented in the whole population till the final stage with at least a small size sub-population. This step also allows a sub-population with small size to recover back to a bigger size if more fit members are generated in sub-sequent generations.

4.2 MGADSM Trajectory Optimization

In this chapter, the objective is to design an interplanetary trajectory mission that consists of m gravity assist maneuvers and n_l deep space maneuvers in each leg of the $m+1$ mission's legs, without a priori knowledge about the swing-by planets or the DSMs. Equation (4.1) shows the total cost of the mission trajectory, Δv_T, that needs to be minimized.

$$\Delta v_T = \|\Delta \mathbf{v}_d\| + \sum_1^m \|\Delta \mathbf{v}_{ps}\| + \sum_1^{m+1} \sum_1^{n_l} \|\Delta \mathbf{v}_{DSM}\| + \|\Delta \mathbf{v}_a\| \tag{4.1}$$

where $\Delta \mathbf{v}_d$ and $\Delta \mathbf{v}_a$ are the departure and arrival impulses, respectively, $\Delta \mathbf{v}_{ps}$ is the post-swing-by impulse of the powered gravity assist only, and $\Delta \mathbf{v}_{DSM}$ is the applied deep space maneuver impulse. The fitness F at a design point, which is maximized to determine the fittest solution, is defined as:

$$F = \frac{1}{\Delta v_T} \tag{4.2}$$

This MGADSM optimization problem is characterized by mixed (continuous and discrete) design variables. The discrete variables determine the planets to swing by (mission scenario) and the direction of flight (posigrade or retrograde). The continuous variables determine the dates and parameters of the maneuvers (swing-bys and DSMs). The fact that some of the design variables are discrete suggests the use of genetic algorithms (22, 23). Each design point (member) in a GA population represents a set of independent design variables. Table 4.1 shows all design variables. The Matlab genetic algorithm toolbox is used to develop the DSMPGA tool. The GA Matlab toolbox handles only problems of continuous design variables. In this MGADSM problem, the discrete variables are modeled as continuous variables and then rounded to the nearest integer, so that the Matlab GA toolbox can still be used.

Table 4.1
The discrete and continuous independent design variables of the MGADSM problem

Discrete Design Variables (DDV)	**Continuous Design Variables (CDV)**
No. of swing-by maneuvers, m	Departure date, t_d
Swing-by planets, P_1,..., P_i	Arrival date, t_a
The count of DSMs in each leg, n_1,..., n_{i+1}	Time of flight, T_1,..., T_i
Flight direction, f	Normalized pericenter altitudes, $\overline{h}_1$,..., $\overline{h}_i$
	Rotation angles, η_1,..., η_i
	Epochs of DSMs, ε_1,..., ε_j
	DSMs, $\Delta \mathbf{v}_1$,..., $\Delta \mathbf{v}_k$

In Table 4.1, i is the maximum possible number of swing-by maneuvers and j is the maximum possible number of total DSMs in the entire trajectory. The term DSM is used to define any thrust impulse applied during the mission course except the launch, arrival, and powered swing-by impulses. The maximum number of independent thrust impulses is k, which can be explained as follows: if n_l is the maximum number of DSMs in a leg, then there are n_l independent thrust impulses if this leg is the first one; while for the consequent legs, the number of required independent thrust impulses is $n_l - 1$. Each thrust impulse $\Delta\mathbf{v}$ consists of three continuous design variables. The epoch of a DSM, ε, specifies the time at which the impulsive maneuver is applied, as a fraction of the associated leg transfer time. For most applications, a DSM could be applied from 10% up to 90% of the total time of flight in the associated leg. Each variable P_l determines the planet to swing by. The range of the discrete variable P_l is 1 through 8, which are the indices for the planets in the solar system, starting from Mercury to Neptune. The order of the variables P_l determines the order of the swing-bys, i.e. the mission scenario. The dates of the mission maneuvers are specified by selecting the departure and arrival dates and the time of flight of each leg except the last one. The normalized pericenter altitudes $\overline{h}$ and the rotation angles η are used to specify the swing-by maneuvers. The flight direction f of the whole mission is either retrograde or posigrade.

For a MGADSM problem, the total number of independent design variables depends on the number of swing-by maneuvers, m, and the number of DSMs in every leg, $n_1, ..., n_{i+1}$. Selecting different values for these discrete design variables changes the mission scenario and the DSM structure of the whole trajectory. Thus, changing the mission scenario and/or the DSM structure is accompanied by a change in the length of the chromosome. This type of problems can not be handled with the standard genetic algorithm because of the variation of strings lengths within the same population. The DSMPGA method is implemented.

For the MGADSM problem, each sub-population represents a single DSM structure with a fixed length mission scenario. This means that, for each sub-population, the number of DSMs in each leg , as well as the number of swing-by maneuvers, are constants. The chromosome length L_s of a specific sub-population is calculated as follow:

$$L_s = 5 + 3m_s + j_s + 2(m_s - z_s) + 3k_s \tag{4.3}$$

where z is the number of swing-by maneuvers which are followed by a zero-DSM leg, and the subscript s is an index for the sub-population. The number of sub-populations depends

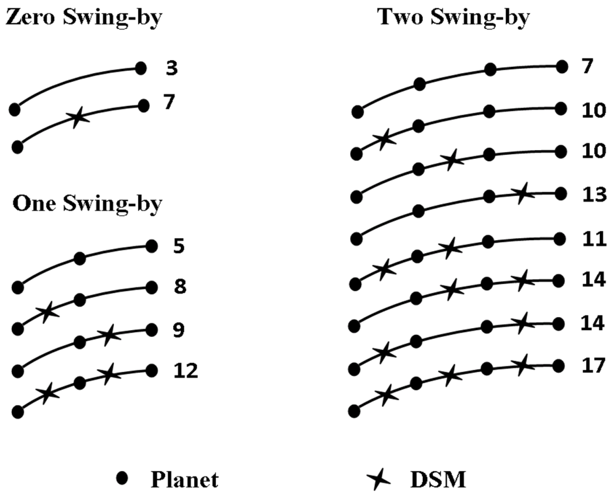

Figure 4.2: DSM structure for different scenarios in DSMPGA tool, and the corresponding number of independent design variables

on the lower and upper bounds of the key design variables m and n_l. For a given value of m, all possible combinations of DSM structure are considered, as shown in Figure 4.2. For instance, if $m = 1$, then the trajectory consists of two legs. Suppose the lower and upper bounds of n_l for each leg are zero and one, respectively. Therefore, there are four possible combinations (sub-populations) of the DSM structure. The four sub-populations in this case are: zero-DSM in both legs, a single DSM in the first leg with zero-DSM in the second leg, a single DSM in the second leg with zero-DSM in the first leg, and a single DSM in each leg. These four sub-populations have different chromosome lengths, as shown in Figure 4.2.

The independent design variables m and n_l are not presented in the chromosome. The bounds of these two variables specify the number of sub-populations, and hence determine the problem size. The DSMPGA method benefits from a formulation for the N-impulse orbit transfer that integrates evolutionary algorithms and a Lambert's problem formalism. This integration of Lambert's problem significantly reduces the size of the design space to include only candidate solutions which satisfy Lambert's problem solution (17, 19).

The DSMPGA is tested for several interplanetary missions ranging from simple to complex missions. The DSMPGA finds the mission scenario (the number of swing-bys and the planets to swing by) as well as the rest of the independent design variables: the times of swing-by, the number of DSMs, the times of DSMs, the magnitudes/directions of DSMs, and the departure/arrival dates. The size of the design space is determined by the bounds of the independent design variables. The number of revolutions in each leg is also optimized by the DSMPGA tool. The time of flight (TOF) of each leg and the time of applying each DSM are selected as design variables to allow for multi-revolution transfers. Lambert's problem is solved once in each leg. Based on the position vectors and the TOF, Lambert's solution may have both single and multi-revolution transfers. The criterion used to select a single or a multi-revolution transfer is as follows: the selected lambert's transfer should minimize the former maneuver cost. This maneuver could be a departure impulse, a DSM, or a swing-by maneuver. In the final leg, the selected lambert's transfer will minimize the former maneuver cost plus the arrival impulse cost.

The main operations in the standard GA are coding, evaluation, selection, crossover and mutation. A scattered crossover operation is conducted with a probability varied from 0.8 to 0.9. An adaptive feasible mutation function is considered so that the design variables bounds are satisfied. Rank scaling function is applied. Roulette wheel method is used in the selection operation. Wider initial ranges are used in order to increase the diversity in the population. The solution obtained by the DSMPGA is not necessary an optimal solution, nor at a local minimum. Therefore, a constrained nonlinear optimization technique is used to improve the solution by finding the closest local minimum to that solution. The local optimizer is only optimizing over the continuous design variables, not the discrete variables. The DSMPGA fittest solution is used as an initial guess in the local search algorithm.

4.3 Numerical Results

This section presents a number of case studies of trajectory optimization for some known interplanetary space missions. Comparisons to other solutions in the literature are presented in Section 4.4.

4.3.1 Earth-Mars Mission

A MGADSM trajectory is optimized for the Earth-Mars mission (EM). The lower and upper bounds, for all design variables, are listed in Table 4.2. The maximum possible number of swing-by maneuvers is selected to be two. The number of DSMs in each leg is selected to be either zero or one impulse. As can be seen from Table 4.2, a gravity

Table 4.2
Bounds of design variables for Earth-Mars mission

Design Variables	Lower Bound	Upper Bound
No. of swing-by maneuvers, m	0	2
Swing-by planets identification numbers, P_1,..., P_i	1 (Mercury)	8 (Neptune)
No. of DSMs in each mission's leg, n_1,..., n_{i+1}	0	1
Flight direction, f	Posigrade	Retrograde
Departure date, t_d	01-Jun-2004	01-Jul-2004
Arrival date, t_a	01-Apr-2005	01-Jul-2005
Time of flight (days)/leg, T_1,..., T_i	40	300
Swing-by normalized pericenter altitude, $\overline{h}_1$,..., $\overline{h}_i$	0.1	10
Swing-by plane rotation angle (rad), η_1,..., η_i	0	2π
Epoch of DSM, ε_1,..., ε_j	0.1	0.9
DSM (km/s), Δv_1,..., Δv_k	-5	5

Table 4.3
MGADSM solution trajectory for the EVM mission using DSMPGA

Mission Parameter	MGADSM Scenario
Departure date,t_d	01-Jun-2004 00:00:00
Departure impulse (km/s)	4.386
DSM date	02-Sep-2004 17:06:11
DSM impulse (km/s)	0.296
Venus Swing-by date	17-Nov-2004 13:51:17
Pericenter altitude (km)	7701.2
DSM date	30-Apr-2005 18:11:07
DSM impulse (km/s)	0.626
Arrival date	19-May-2005 01:47:07
Arrival impulse (km/s)	5.392
Time of flight (days)	169.58 , 182.49
Mission duration (days)	352.07
Mission Cost (km/s)	**10.7**

assist maneuver could be performed with any planet in the solar system, from Mercury to Neptune. The direction of flight can be posigrade or retrograde. The time of flight of each leg, except the last one, is selected between 40 and 300 days.

The population is divided into 14 sub-populations (2 for zero swing-by, 4 for one swing-by, and 8 for two swing-by maneuvers). The maximum number of design variables is 17 which is the design space size in the last sub-population (two swing-by with a single DSM in each leg). The other sub-populations have different sizes for the design space. The proportionality constant for initial population size, c, is selected to be 10. The overall population size is 1400. The number of generations in each stage is selected to be 30, and the algorithms stops after 5 stages. The elite count is selected to be 10%, with a minimum

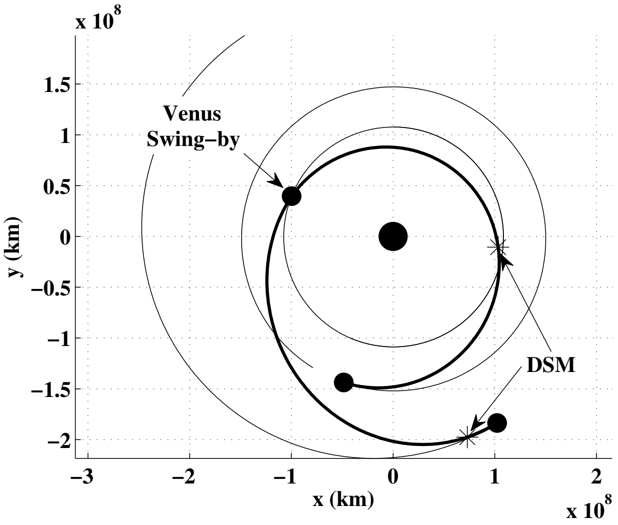

Figure 4.3: Optimal EVM mission using DSMPGA optimization tool.

of 10 individuals in each sub-population. A local optimizer uses the fittest GA solution as an initial guess to find a local minimum in its neighborhood. The resulting solution has a single swing-by maneuver at Venus, with a total cost of 10.703 km/s. The fittest trajectory has a single DSM in each leg as shown in Table 4.3. Figure 4.3 shows the obtained EVM trajectory.

4.3.2 Earth-Jupiter Mission

The DSMPGA tool is used to investigate the Earth-Jupiter mission. The objective is to design an optimal MGADSM trajectory with minimum propellent consumption. Wide ranges of all design variables are allowed in the optimization to explore more of the design space. The design variables' bounds are listed in Table 4.4. The population is divided into 14 sub-populations. The maximum number of design variables is 17 which is used in the last sub-population (two swing-by with a single DSM in each leg). The other sub-populations have different numbers and combinations of the independent design variables. The proportionality constant for initial population size, c, is selected to be 20. The overall population size is 2800. The number of generations in each stage is selected to be 40, and the algorithms stops after 5 stages. The elite count is selected to be 10%, with a minimum of 10 individuals in each sub-population. A local optimizer uses the fittest GA solution as an initial guess to find a local minimum.

The resulting fittest scenario is a two swing-by trajectory with swing-bys around Venus then Earth (EVEJ). The solution is a posigrade multi-revolution trajectory. The trajectory has a single DSM in the second leg (VE) with a total cost of 10.125 km/s, as shown in Table 4.5. The optimal EVEJ trajectory is shown in Figure 4.4. The DSM amplitude is 2.14 m/s which seems an insignificant value with respect to the total trajectory cost. It is expected that this small DSM may be vanished with more iterations. A powered swing-by maneuver is obtained with the second swing-by planet. The post-swing-by impulse is 0.443 km/s applied during the Earth swing-by.

Table 4.4
Bounds of Earth-Jupiter mission's design variables

Design Variables	**Lower Bound**	**Upper Bound**
No. of swing-by maneuvers, m	0	2
Swing-by planets identification numbers, P_1,..., P_i	1 (Mercury)	8 (Neptune)
No. of DSMs in each mission's leg, n_1,..., n_{i+1}	0	1
Flight direction, f	Posigrade	Retrograde
Departure date, t_d	01-Sep-2016	30-Sep-2016
Arrival date, t_a	01-Sep-2021	31-Dec-2021
Time of flight (days)/leg, T_1,..., T_i	80	800
Swing-by normalized pericenter altitude, $\overline{h}_1$,..., $\overline{h}_i$	0.1	10
Swing-by plane rotation angle (rad), η_1,..., η_i	0	2π
Epoch of DSM, ε_1,..., ε_j	0.1	0.9
DSM (km/s), Δv_1,..., Δv_k	-5	5

Table 4.5
MGADSM solution trajectory for the EVEJ mission using DSMPGA

Mission Parameter	**MGADSM Scenario**
Departure date,t_d	01-Sep-2016 01:54:21
Departure impulse (km/s)	3.487
Venus Swing-by date	05-Sep-2017 10:21:59
Pericenter altitude (km)	1339.21
DSM date	25-May-2018 09:03:27
DSM impulse (km/s)	0.00214
Earth Swing-by date	30-Mar-2019 06:01:26
Post-swing-by impulse (km/s)	0.443
Pericenter altitude (km)	637.8
Arrival date	29-Sep-2021 00:31:15
Arrival impulse (km/s)	6.193
Time of flight (days)	369.35 , 570.82 , 913.77
Mission duration (days)	1853.94
Mission Cost (km/s)	**10.125**

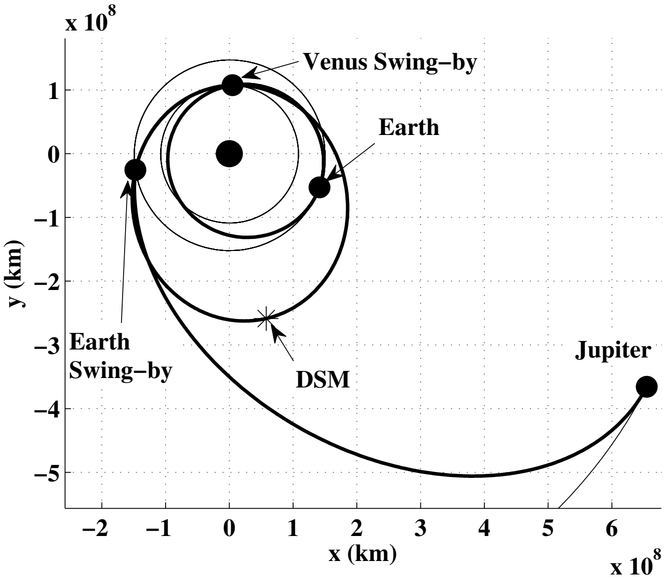

Figure 4.4: Optimal EVEJ mission using DSMPGA optimization tool.

4.3.3 Messenger Mission (Easy Version)

Messenger is the first mission to explore the planet Mercury. Messenger trajectory follows a path through the inner solar system to rendezvous Mercury. The easy version is considered to simplify the problem structure by excluding the resonant swing-bys around planet Mercury (55). The DSMPGA tool is used to design an optimal MGADSM trajectory of the rendezvous mission to Mercury. The design variables' bounds are listed in Table 4.6. These values are selected to be consistent with the problem database listed in Reference (55).

Table 4.6
Bounds of Messenger (easy version) mission's design variables

Design Variables	**Lower Bound**	**Upper Bound**
No. of swing-by maneuvers, m	1	3
Swing-by planets identification numbers, P_1,..., P_i	2 (Venus)	4 (Mars)
No. of DSMs in each mission's leg, n_1,..., n_{i+1}	0	1
Flight direction, f	Posigrade	Retrograde
Departure date, t_d	01-Jan-2003	31-Mar-2003
Arrival date, t_a	01-Jan-2006	30-Jun-2006
Time of flight (days)/leg, T_1,..., T_i	30	400
Swing-by normalized pericenter altitude, $\overline{h}_1$,..., $\overline{h}_i$	0.1	5
Swing-by plane rotation angle (rad), η_1,..., η_i	0	2π
Epoch of DSM, ε_1,..., ε_j	0.1	0.9
DSM (km/s), Δv_1,..., Δv_k	-5	5

Table 4.7
Optimal MGADSM trajectory of Messenger mission (easy version) using DSMPGA

Mission Parameter	First Scenario	Second Scenario
Departure date	17-Mar-2003 00:02:43	14-Feb-2003 08:13:21
Departure impulse (km/s)	1.421	1.002
DSM date	20-Jun-2003 08:04:26	05-Jul-2003 10:56:08
DSM impulse (km/s)	0.904	0.906
Earth Swing-by date	20-Apr-2004 00:02:43	22-Apr-2004 10:36:27
Post-swing-by impulse (km/s)	0.00011	-
Pericenter altitude (km)	4749.47	4155.028
DSM date	-	27-May-2004 00:20:53
DSM impulse (km/s)	-	0.00024
Venus Swing-by date	15-Oct-2004 22:58:49	18-Oct-2004 07:47:17
Pericenter altitude (km)	12251.66	11395.952
DSM date	20-Jun-2005 02:59:57	04-Mar-2005 05:01:11
DSM impulse (km/s)	0.258	0.192
Venus Swing-by date	11-Aug-2005 04:39:48	13-Aug-2005 15:36:22
Pericenter altitude (km)	605.2	605.2
DSM date	05-Oct-2005 22:29:27	07-Oct-2005 07:00:37
DSM impulse (km/s)	1.448	1.514
Arrival date	08-Feb-2006 03:20:21	08-Feb-2006 15:46:58
Arrival impulse (km/s)	4.6	4.589
Time of flight (days)	400, 178.96, 299.24, 180.94	433.1, 178.88, 299.33, 179.01
Mission duration (days)	1059.14	1090.32
Mission Cost (km/s)	**8.6312**	**8.203**

There are 28 sub-populations with different DSM structure. The maximum number of DSMs is four while the minimum number of DSMs is zero. The maximum number of design variables is 22 which is used in the last sub-population (three swing-by with a single DSM in each leg). The other sub-populations have different numbers and combinations of the independent design variables. The proportionality constant for initial population size, c, is selected to be 15. The overall population size is 5670. The number of generations in each stage is selected to be 40, and the algorithms stops after 5 stages. The elite count is selected to be 10%, with a minimum of 10 individuals in each sub-population. A local optimizer uses the fittest GA solution as an initial guess to find a local minimum.

The obtained optimal solution is a three swing-by trajectory with the same planet sequence as of the actual Messenger mission (easy version) scenario (EEVVY). The solution is a posigrade trajectory with three DSMs, a single DSM in each leg except the second leg. The obtained solution is listed as the first scenario in Table 4.7. The first scenario solution has a total cost of 8.6312 km/s, and is shown in Figure 4.5. To improve the mission cost, wider bounds are used for the time of flight design variables. The upper bound of the time of flight is changed to 500 days instead of 400 days. Only 8 sub-populations are considered in the second iteration. These sub-populations represent only three swing-by scenarios with a single DSM in the first leg. The other legs could have zero or one DSM. The proportionality constant for initial population size, *c*, is selected to be 30. The overall population size is 4200. The number of generations in each stage is selected to be 40, and the algorithms stops after 5 stages. The obtained solution is listed in Table 4.7 as the second scenario. It has the same planet sequence of the first scenario but with different DSM structure. The optimal trajectory has four DSMs, a single DSM in each leg, with a total cost of 8.203 km/s. The second scenario trajectory is shown in Figure 4.6. In the second scenario, the DSM amplitude in the second leg (VV) is 0.24 m/s which seems an insignificant value with respect to the total trajectory cost. It is expected that this small DSM may be vanished with more iterations.

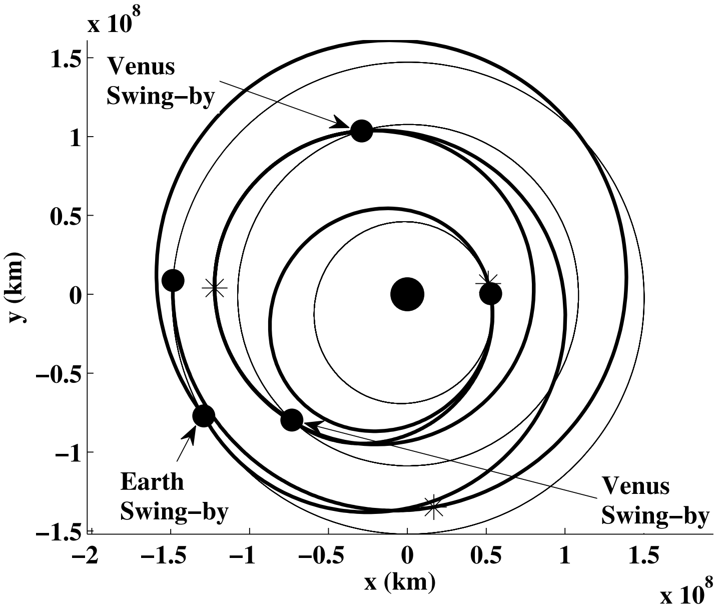

Figure 4.5: Optimal Messenger mission (easy version) using DSMPGA, first scenario (3 DSMs).

4.4 Comparisons and Discussion

The developed DSMPGA has the capability to compute the number of swing-bys and the planets to swing by, in addition to the rest of the classical MGADSM design variables, in the best solution trajectory. Then the mission scenario, the direction of flight, and the

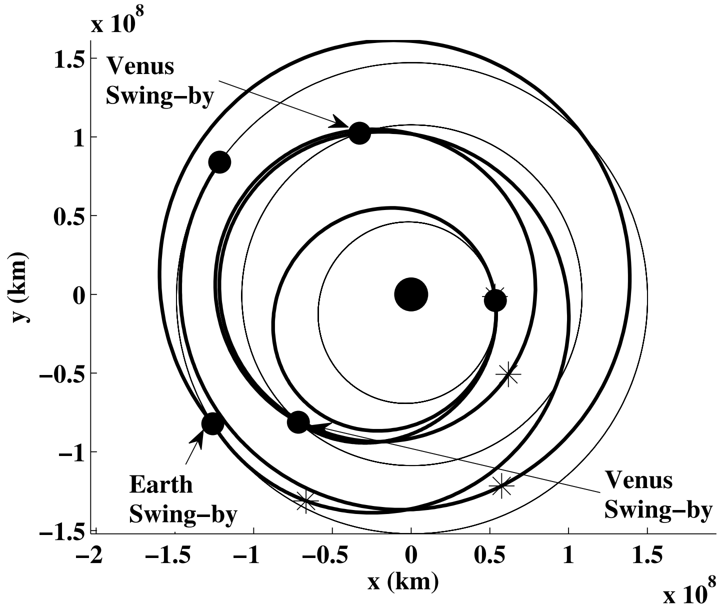

Figure 4.6: Optimal Messenger mission (easy version) using DSMPGA, second scenario (4 DSMs).

number of DSMs are fixed, and the continuous variables are tuned by a local optimizer. The local optimizer implements a constrained nonlinear optimization technique.

The Earth-Venus-Mars mission trajectory optimization has been presented in the literature (15). The extended primer vector theory is used to obtain the solution presented in Reference (15). The trajectory has a single swing-by maneuver and a single DSM. In implementing the primer vector method, fixed departure and arrival dates were assumed with 340 days mission duration. The Venus swing-by time was also constrained to occur at 165 days from departure. The resulting solution has a DSM of 68.7 m/s in the first leg at 96.08 days from mission start date. The total cost of the mission is 10.786 km/s (15). The same problem has been solved using the ITO-HGGA tool (7), and the obtained cost is 10.728 km/s. In (7), the trajectory has a single Venus swing-by and a single DSM (180.1 m/s) in the first leg. In this chapter, the solution obtained using the DSMPGA tool has one swing by Venus and one DSM in each leg, 0.296 and 0.626 km/s, respectively. The total cost of the mission is 10.7 km/s, as shown in Table 4.3. The difference in the total cost obtained using the DSMPGA tool, as compared to that of References (15) and (7), is accompanied by small changes in the mission's departure, swing-by, and arrival dates, without significantly changing the total mission duration.

A minimum-cost solution trajectory for the Earth-Jupiter mission is addressed in Reference (15). A fixed planet sequence EVEJ is assumed. The departure, arrival, and swing-by dates were also assumed fixed, with a launch in 2016 and a mission duration of 1862 days. The primer vector theory solution has four DSMs. Two DSMs are applied in the first two legs. The total transfer cost for this solution is 10.267 km/s. The DSMPGA tool developed in this chapter is able to find automatically the known swing-by sequence, EVEJ, for the Jupiter mission in 2016. As listed in Table 4.5, the optimal MGADSM solution has 10.125 km/s total cost with a single DSM in the whole trajectory. The ITO-HGGA tool was used to solve the same mission in Reference (7). The obtained trajectory from the ITO-HGGA tool has the same planet sequence and DSM structure but with higher total cost, 10.178 km/s (7).

The Messenger mission (easy version) trajectory design problem has been addressed in several studies (16, 55), where it is always assumed that a fixed swing-by sequence (EEVVY) is known. According to the GTOP database, the best solution is found by F. Biscani, M. Rucinski and D.Izzo, using PaGMO, a new version of DiGMO, based on the asynchronous island model (55). The total cost is 8.63 km/s with a single DSM in each leg. Olympio and Izzo (16) recently developed an algorithm to find only the optimal DSM structure in a given trajectory scenario. They used the same ephemeris tool of the GTOP database (55) but with a free number of applied DSMs. For the Messenger planets' sequence described in the GTOP database (55), Reference (16) found a trajectory with a total cost of 8.494 km/s. The DSM structure is 2-0-1-1 in consequent legs respectively. The DSMPGA is used twice, in this chapter, to investigate this mission. In the first iteration, the same design variables' bounds of the GTOP database are used. The obtained optimal solution has a total cost of 8.6312 km/s (as seen in Table 4.7), which is very close to the reported result in the GTOP database (55). However, the DSMPGA tool has the advantage of finding the planets sequence as well as the DSM structure. In the second iteration, a better solution is found with a total cost of 8.203 km/s. The significant improvement in fuel consumption is a result of using wider windows for the time of flight variables.

Figure 4.7 shows the change in the size of each sub-population (number of individuals) in subsequent stages. Eight sub-populations are shown which represent the second solution obtained in Table 4.7) of the Messenger easy mission. The initial size of each sub-population has different number of individuals depending on the number of independent design variables in the sub-population. The size of each sub-population varies over stages

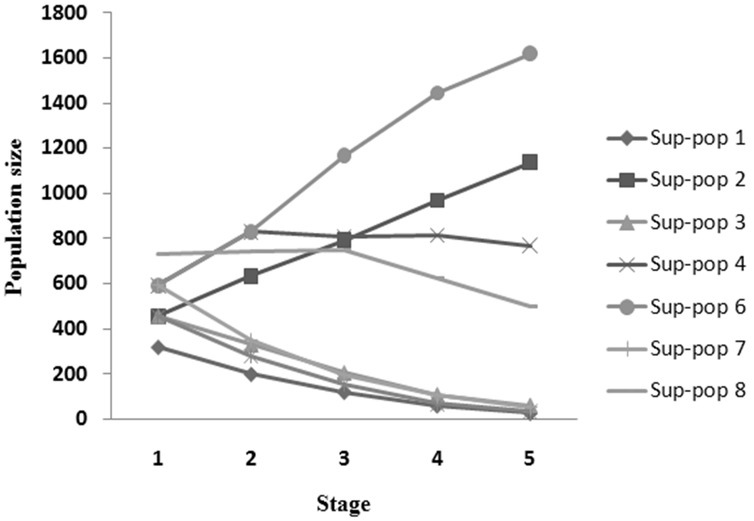

Figure 4.7: Sub-populations size performance for Messenger mission (easy version), second iteration.

depending on the behavior of the fitness function. The higher fit sub-populations increase in size, while the lower fit sub-populations decrease in size.

If a standard GA is implemented to solve the MGADSM problem presented in this chapter, considering all the design space, then the implementation would be as follows. A population will be constructed for each possible design space size, e.g. eight populations in the Messenger easy mission in Figure 4.7. Standard GA is then implemented for each population till a stopping criterion is satisfied. The sizes of all populations will not change, and there will be no stages to stop at during the iterations process. So, this is equivalent to solving eight optimization problems in the case of the Messenger easy mission. The DSMPGA method presented in this chapter is computationally more efficient compared to a standard GA. Only one population is processed, the size of which is much less than the summation of the eight populations processed by a standard GA.

The hidden genes genetic algorithm (HGGA) (7) also solves the VSDS problems. This method was implemented to solve the MGADSM problem in (7). The hidden genes concept creates only one population with all solutions of variable length represented in the population. The HGGA has a method of processing variable-length chromosomes; parents of different lengths can be used to generate offsprings of different lengths from their parents. Unlike the hidden genes concept, the DSMPGA concept has no genes exchange between the chromosomes of different lengths. All chromosomes lengths are represented, however, in the DSMPGA, and those that have more fit solutions increase in the number of members in subsequent generations. In a sense, we can state that the DSMPGA is more suitable for VSDS problems where the number of different chromosomes lengths is small, whereas the HGGA method is more suitable when the number of different chromosomes lengths is higher.

4.5 Summary

A new technique to solve the optimization problem of MGADSM trajectory is introduced in this chapter. A novel dynamic-size multiple population genetic algorithm is developed to handle optimization problems with variable size design space. Sub-populations, each has members of same size design space, are constructed. A standard GA is applied for a number of stages; each stage is a number of consequent generations in which the size of the sub-population remains fixed. At the end of each stage, the sizes of all sub-populations are changed based on the fitness of the members in each individual sub-population. This algorithm leads to the increase in the size of the sub-populations of more fit members. The DSMPGA can determine, without a priori knowledge, the number of swing-by and DSM maneuvers, the planets to swing by, and the times of swing-bys and DSMs, in addition to the rest of the design variables. The solutions found using the developed tool match the best known solutions for the problems presented in this chapter.

Chapter 5

Solution of GTOC Using HGGA Concept

The objective of this chapter is to introduce a preliminary approach to solve the low-thrust propulsion missions. The optimization problem is considered for trajectories of a spacecraft rendezvous mission to a group of asteroids with revisiting. The problem of low-thrust trajectory optimization is usually formulated to minimize the propellent consumption. Hence, the greatest number of asteroids could be visited within the pre-specified mission duration. The spacecraft is launched from the Earth. The departure date is selected from the preliminarily set period of time. The visited asteroids are chosen from a group of asteroids. The spacecraft should visit each astroid twice, the first time is a rendezvous while the second is a penetration (flyby). A two-body dynamics model is considered as a framework for the developed algorithm. Therefore, Keplerian orbits are assumed to describe the movement of the Earth and asteroids around the Sun. The only forces acting on the spacecraft are the Sun's gravity and the thrust produced by the engine.

The problem is solved in four stages, as illustrated in Figure 5.1. At the first stage, an impulsive trajectory problem is solved to determine a preliminary mission scenario. The scenario of the mission refers to the number and sequence of the visited asteroids. The segment between any two consequent asteroids is called a leg. The time of flight (TOF) of each leg is also determined in this stage. Two-impulse transfer problem is considered for each leg. A Lambert's problem is solved to find the impulsive preliminary transfer orbit. The hidden genes genetic algorithm (HGGA) tool is used as an optimization technique (7).

The second stage is used to provide an initial guess for the continuous thrust trajectories. For each leg in the trajectory, the initial and final asteroids are determined from the first stage as well as the TOF. Standard genetic algorithm is used to calculate a poor continuous thrust trajectory for each leg. This trajectory is used as an initial guess for the next stage. An optimum control module is then utilized in the third stage to calculate accurate continuous thrust trajectories. General pseudospectral optimal control software (GPOPS) is used in this stage. The obtained solution from the GPOPS module is a visible continuous thrust trajectory but with a limited number of nodes. This solution is then used as an initial guess in the final stage to compute the final detailed trajectory. In the forth stage, a constrained nonlinear optimization technique is used to provide the optimal trajectory data at one-day increments for each leg of the trajectory.

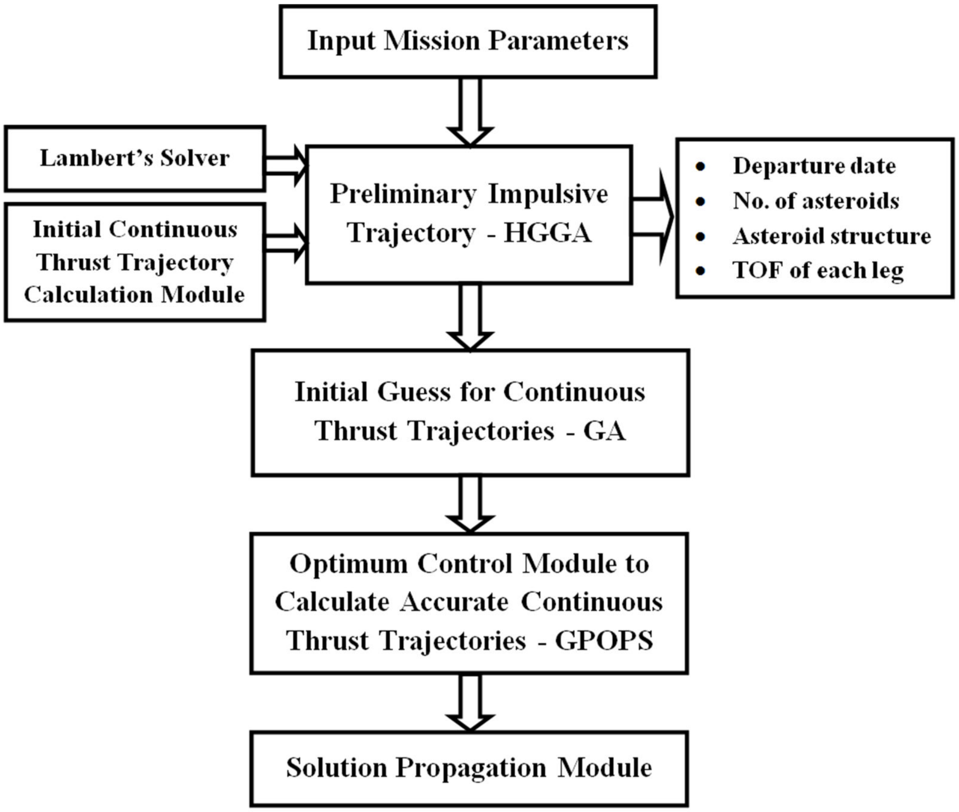
Input Mission Parameters
Lambert's Solver
Initial Continuous Thrust Trajectory Calculation Module
Preliminary Impulsive Trajectory - HGGA
• Departure date
• No. of asteroids
• Asteroid structure
• TOF of each leg
Initial Guess for Continuous Thrust Trajectories - GA
Optimum Control Module to Calculate Accurate Continuous Thrust Trajectories - GPOPS
Solution Propagation Module

Figure 5.1: A flowchart shows the consequent stages used to solve the problem.

5.1 Preliminary Impulsive Trajectory

The problem of the first stage is formulated as an impulsive transfer optimal trajectory. Let m is the number of visited asteroids. So, the mission has $m+1$ different legs in the trajectory. The dates of asteroids visiting are determined by specifying the time of flight for each leg. Time window is specified for each leg TOF. The arrival date could be calculated from the combined TOFs of the visited asteroids. Since the total mission duration and total propellent mass are limited, the number of visited asteroids is also limited. The calculations of the dependent variables are carried out starting from the departure planet (Earth), and from one leg to the next and so on. In any trajectory leg, the spacecraft trajectory is solved as a two-impulse trajectory problem. There are only two variables (20), the initial and final dates of the related leg. For a candidate solution, the leg time of flight T_l could be computed. The transfer orbit is then calculated by solving a Lambert's problem. The spacecraft initial and final impulses are then calculated.

The asteroid maneuver type could be rendezvous or penetration. A rendezvous maneuver requires the spacecraft position and velocity to be the same as those of the target asteroid. A penetration maneuver requires concurrence of position of spacecraft and a target asteroid. During a penetration maneuver, the spacecraft relative velocity (w.r.t the target asteroid) should not less than a pre-specified minimum value $\Delta v_{p_{min}}$. A revisiting mission is studied in this chapter. First, the spacecraft should perform a rendezvous maneuver with the asteroid to deliver a scientific equipment. In the second maneuver with the same asteroid, the spacecraft should flyby (penetrate) through the astroid to deliver a penetrator. The penetration maneuver is not accepted before delivering the scientific equipment (rendezvous). The astroid is not allowed to be visited for a third time. The spacecraft has a fixed initial mass which is not affected by the launch exceed velocity.

For the first leg, the target astroid is a rendezvous. The required initial and final impulses, $\Delta \mathbf{v}_i$ and $\Delta \mathbf{v}_f$, are calculated based on Lambert's solution. The initial impulse of the first leg is provided through the launcher. The on-board propellent provides the rendezvous impulse of the first asteroid. For the consequent asteroids, the analysis of each leg is determined based on the maneuver type of the target astroid. For a rendezvous target astroid, the rendezvous impulse is calculated from Lambert's problem. In case of a flyby target astroid, the spacecraft velocity vector (magnitude and direction) does not change over the penetration. An initial impulse is required for any leg to guarantee that the spacecraft final position is exactly the same as the target astroid. Therefore, the cost of each leg depends on the required applied impulse(s). The total cost of the impulsive trajectory is the summation of the cost of the involved legs. The multi-revolution transfers are considered in the analysis through choosing the time of flight of each leg as as a design variable. Lambert's problem is solved once in each leg. Based on the TOF and the position vectors , Lambert's solution may have both single and multi-revolution transfers. The selection criterion is as follows: the selected lambert's transfer should minimize the impulsive maneuver cost of the related leg.

The impulsive problem is formulated as follows: For a given range for departure dates from the Earth, find the optimal selections for the number of asteroids, the asteroids to visit, the visiting type (rendezvous or penetration), the dates of visiting, and the exact launch and arrival dates. An optimization problem is formulated to minimize the total impulsive mission cost and maximize the number of visited asteroids. The problem is characterized by solution-dependent design variables. This means that different solutions have different number of design variables. The number of required design variables depends on the actual visited asteroids in each solution. The number of visited asteroids is computed based on the mission requirements (mission duration and propellent mass). Therefore, standard genetic algorithm could not be implemented as an optimization technique. To overcome this problem, the concept of hidden genes genetic algorithm (HGGA) is used (7). This concept is capable to handle all solutions in the design space as if they all have the same chromosome length, and hence allows the implementation of standard genetic operations.

In a general solution, some of the variables in the chromosome will be ineffective in cost function evaluation. In the HGGA tool, the genes describing these variables are called hidden genes. The hidden genes, however, will be used in the genetic operations in generating future generations. The independent design variables to be optimized in the impulsive problem are selected as follows: departure date (t_d), time of flight of each leg (T_1, T_2, ..., T_n), and the asteroids to be visited (P_1, P_2, ..., P_n); where n is the maximum number of targeted asteroids (specified by the user). Each member in the design space has a fixed length L, where $L = 1 + 2n$. Some of the variables in the member are ineffective variables (hidden genes). The number of effective variables is varied from one member to another. Figure 5.2 shows typical solution chromosomes where the variables at the top of the figure are the independent design variables. Each row in Figure 5.2 represents a member (single individual) in the GA population.

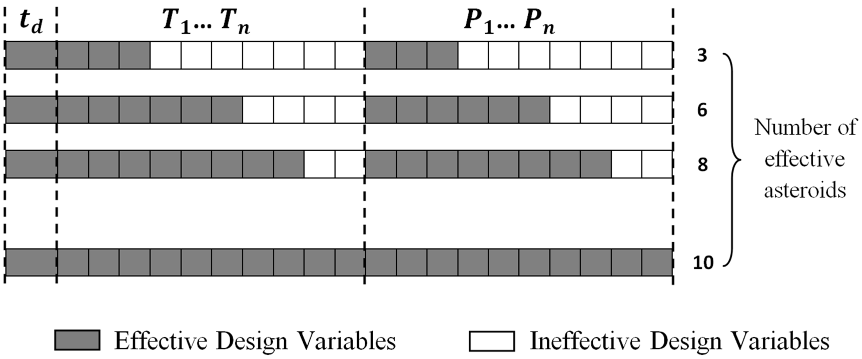
t_d
$T_1 \dots T_n$
$P_1 \dots P_n$
3
6
8
10
Number of effective asteroids
Effective Design Variables
Ineffective Design Variables

Figure 5.2: Typical chromosomes for the trajectory optimization problem used in GTOC problem; each row represents a chromosome.

The HGGA tool has the capability to handle two categories of design variables (discrete and continuous). The variables t_d, T_1, ..., T_n are coded as CDVs. While the DDVs are used to code the asteroids identification numbers P_1, ..., P_n. Design variables coding is performed as described in Section 3.1. Given that the maximum possible number of astroid maneuvers is n, then n discrete design variables P_1, P_2, ..., P_n are created. Each variable P_l determines the astroid about which the l^{th} maneuver occurs, where the maneuver could be a rendezvous or penetration. The range of the discrete variable P_l depends on the size of the asteroids set. The order of the asteroids is the same as the order of the variables P_l in the chromosome. The sequence of the visited asteroids determines the mission scenario. For each member in the design space, the analysis of the consequent legs is performed to compute the mission cost and the propellent consumption. The calculations are stopped for two reasons. The first is the total mission duration is accomplished, while the second is the propellent mass is over. Therefore, the actual visited asteroids m could be less than the proposed visited asteroids n. In this case, there are $2m+1$ genes are active in the chromosome which represent the effective design variables (t_d, T_1, ..., T_m, P_1, ..., P_m). Hence, the ineffective (hidden) genes in the chromosome are $2(n-m)$ variables.

In this stage, the objective is to minimize the total impulsive cost Δv_T of the trajectory and maximize the number of visited asteroids with revisiting priority. The astroid revisit mission is defined as: first the spacecraft should rendezvous with an astroid, then a penetration flyby is performed with the same astroid. The fitness function of the first stage F_1 at a design point, which is minimized to determine the fittest impulsive solution, is defined as,

$$F_1 = (\frac{f_1}{J})^{f_2} + (\Delta v_T)^{f_3} \tag{5.1}$$

where f_1, f_2, and f_3 are the fitness weight parameters. J is the performance index of the astroid revisit mission. The performance index is calculated as follows,

$$J = \sum_{i=1}^{m} (\alpha_i + \beta_i) \tag{5.2}$$

where m is the total number of visited asteroids in the mission, α_i is the rendezvous maneuver coefficient, and β_i is the penetration maneuver coefficient. The coefficient α_i should be 0.2 if a rendezvous is fulfilled. The coefficient β_i should be 0.8 in case of a successful penetration maneuver. A penetration maneuver is not allowed before a rendezvous maneuver with the same astroid. The spacecraft penetration relative velocity should not exceed 0.4 km/s. The performance index increases the importance of revisiting by giving higher weight for the penetration than the rendezvous maneuver. The total impulsive cost Δv_T is calculated as follows,

$$\Delta v_T = \sum_{1}^{m-1} \|\Delta \mathbf{v}_i\| + \sum_{1}^{k} \|\Delta \mathbf{v}_f\| \tag{5.3}$$

where k is the number of rendezvous maneuvers in the trajectory. The fitness weight parameters are carefully selected to provide a maximum number of asteroid missions, and in the same time, to minimize the propellent consumption. The values of the parameters f_1, f_2, and f_3 are selected to be 100, 2, and 1.5, respectively.

A Matlab toolbox (GENETIC v2.1) is used (53). A uniform crossover operation is used with a probability varied from 0.92 to 0.98. The mutation probability is selected between 0.01 and 0.08. Roulette wheel is used in selection operation. Proportional ranking is implemented in the analysis. To increase the diversity in the population, niching principle is applied by degrading the fitness of the similar individuals (54). A simple niching technique is implemented in the developed impulsive stage. Every ten generations, the fitness are degraded for the current fittest solution, and all other solutions that have a similar astroid scenario to the fittest solution. The solution obtained by the GA is not necessarily an optimal solution, nor is it at a local minimum. Therefore, a constrained nonlinear optimization technique is used to improve the solution by finding the closest local minimum to that solution. The local optimizer is only optimizing over the continuous design variables, not the discrete variables. The genetic algorithm solution is used as an initial guess in the local search algorithm.

5.2 Initial Continuous Thrust Trajectory

In the second stage, an initial guess for the continuous thrust trajectory is introduced. The problem is separately solved for each leg of the optimal scenario obtained from the first stage. The input parameters for the second stage are obtained from the fittest solution in the first stage. These parameters are the initial and final asteroids, their maneuver type (rendezvous or penetration), and the date of each maneuver. The initial and final spacecraft position vectors could be calculated from that parameters as well as the TOF of leg. The initial and final spacecraft velocity vectors are determined based on the maneuver type. If the initial or final asteroid is associated with a rendezvous maneuver, the spacecraft velocity vector should be the same as the velocity vector of the related asteroid. In a penetration maneuver, there are two cases. First, if the initial asteroid has a penetration, the continuity principle is applied and the spacecraft velocity vector should be the same as the final spacecraft velocity of the former leg. This velocity is the actual spacecraft velocity which is determined after solving the former leg by the developed optimum control module. Second, the final astroid has a penetration maneuver. In this case, the spacecraft velocity obtained from Lambert's solution of the current leg is considered as an acceptable initial guess. This vector is then accurately calculated in the next stage.

The spacecraft has a bounded thrust level magnitude T_c. The thrust direction is not constrained. The applied continuous thrust vector $\mathbf{T}_c$ during thrusting periods is calculated as follows,

$$\mathbf{T}_c = T_c \begin{pmatrix} \cos\delta\cos\alpha \\ \sin\delta\cos\alpha \\ \sin\delta \end{pmatrix} \tag{5.4}$$

where δ and α are the rotation angles which define the thrust direction in the inertial frame. The leg is divided into N_s segments. The number of segments depends on the TOF of the leg. Higher TOF requires a higher number of segments. The required continuous thrust vector is considered constant over each segment, as shown in Figure 5.3. The thrust level is limited with a maximum value $T_{c_{max}}$. The rotation angles δ and α are bounded as follows, $-\pi/2 \leq \delta \leq \pi/2$ and $0 \leq \alpha \leq 2\pi$, respectively.

The motion of the spacecraft around the Sun is governed by the following equations:

$$\dot{x} = v_x, \quad \dot{y} = v_y, \quad \dot{z} = v_z, \quad \dot{m} = \frac{-T_c}{I_{sp} g_E} \tag{5.5}$$

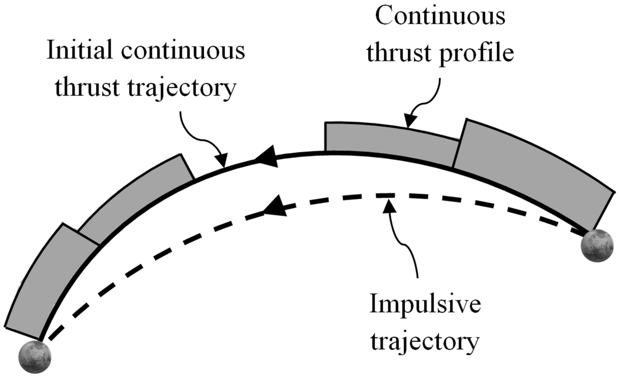

Figure 5.3: Initial continuous thrust trajectory vs. impulsive trajectory.

$$\dot{v}_x = -\frac{\mu_S x}{r^3} + \frac{T_{c_x}}{m}, \quad \dot{v}_y = -\frac{\mu_S y}{r^3} + \frac{T_{c_y}}{m}, \quad \dot{v}_z = -\frac{\mu_S z}{r^3} + \frac{T_{c_z}}{m} \tag{5.6}$$

where x,y,z are the spacecraft position components, v_x,v_y,v_z are the spacecraft velocity components, g_E is the standard acceleration due to gravity on the Earth surface (9.80665 m/s^2), I_{sp} is the specific impulse of the onboard engine, T_{c_x},T_{c_y},T_{c_z} are the thrust components, m is the spacecraft mass, μ_S is the Sun gravitational constant, and r is the distance from the Sun. For a single point in the design space, the thrust components are calculated from Equation 5.4. The spacecraft initial state (position and velocity) is determined from the former stage as well as the TOF of the current leg. Then, the spacecraft is propagated using the dynamic model described in Equations 5.5 and 5.6. The final spacecraft position and velocity vectors, $\mathbf{r}_f$ and $\mathbf{v}_f$, are calculated. The errors in spacecraft final position and velocity, Δr and Δv, are determined as follows,

$$\Delta r = \|\mathbf{r}_f - \mathbf{r}_{req}\|, \quad \Delta v = \|\mathbf{v}_f - \mathbf{v}_{req}\| \tag{5.7}$$

where $\mathbf{r}_{req}$ and $\mathbf{v}_{req}$ are the required spacecraft final position and velocity vectors.

An optimization problem is formulated to calculate the approximated thrust vector over the leg trajectory. The objective is to minimize the propellent mass consumption m_p of the current leg as well as the errors in the spacecraft final position and velocity. The fitness function of the second stage F_2 at a design point, which is minimized to determine the fittest thrust vector, is defined as,

$$F_2 = m_p + (\frac{\Delta r}{1000} - 1)^2 + (\frac{\Delta v}{0.001} - 1)^2 \tag{5.8}$$

Standard genetic algorithm is used as an optimization technique to solve this stage. The independent design variables are T_c, δ and α for each single segment in the trajectory leg. Since there are N_s segments, the total number of independent design variables is $3N_s$. The genetic algorithm toolbox developed by Matlab optimization tool is used. Continuous design variable coding is implemented. The lower and upper bounds of each variable are considered. A scattered crossover operation is conducted with a probability varied from 0.8 to 0.9. An adaptive feasible mutation function is considered so that the design variables bounds are satisfied. Rank scaling function is applied. Roulette wheel is used in selection operation. The population size and the number of generations are problem dependent parameters. They are selected based on how complex is the trajectory. A constrained nonlinear optimization technique is then used to improve the solution. The genetic algorithm fittest solution is used as an initial guess in the local search algorithm. The local optimizer should find the closest local minimum to the fittest solution obtained by GA.

5.3 GPOPS

An optimum control module is then utilized in the third stage to calculate accurate continuous thrust trajectories. General pseudospectral optimal control software (GPOPS) is used in this stage. The obtained solution from the GPOPS module is a visible continuous thrust trajectory but with a limited number of nodes. This solution is then used as an initial guess in the final stage to compute the final detailed trajectory.

5.4 Thrust Profile Scheduling

In this stage, it is required to compute a detailed trajectory data at one-day increments for each leg in the trajectory. The trajectory data includes the spacecraft position and velocity vectors, thrust vector, spacecraft mass, and the time of applying the thrust. The obtained thrust profile from the third stage has a limited number of nodes. The node is the point where the continuous thrust vector is applied. The thrust vector is not provided in between these nodes. The calculated thrust vector at each node could not be fixed over the trajectory till the consequent node. Therefore, a scheduling technique should be applied to determine the required thrust profile along the trajectory as well as the rest of the spacecraft trajectory data. The Lagrangian interpolation is used to calculate the magnitude and direction of the thrust vector at one-day increments. The interpolated thrust profile usually violates the thrust level constraint. Hence, a correction method should be applied to provide an acceptable thrust profile. The final thrust profile should provide the trajectory constraints as well as the thrust level limitation. For each leg, the errors in the spacecraft final position and velocity should not exceed the constraints. These constraints are 1000 km for the position and 1 m/s for the relative velocity in case of a rendezvous transfer.

A constrained nonlinear optimization technique is used to obtain the final trajectory data. The local optimizer is separately applied for each leg taking into account the continuity principle. The independent design variables are the thrust components at each one-day increment. Therefore, the total number of design variables is proportional with the leg TOF. The initial input is the interpolated thrust profile which is calculated from the GPOPS solution. The same dynamic model described in the second stage is considered. The fitness function F_2, stated in Equation 5.9, is also used in the local optimizer. The objective is to minimize the propellent mass and to provide the position and velocity constraints.

5.5 Numerical Results

The problem of the 5^{th} global trajectory optimization competition (GTOC5) is solved (55). The departure date must lie in the range from 2015 to 2025. The spacecraft has a fixed initial mass, i.e. wet mass, $m_i = 4000$ kg. The spacecraft dry mass is $m_d \geq 500$ kg. The spacecraft should deliver a 40 kg scientific equipment at each rendezvous maneuver and a 1 kg penetrator at each flyby maneuver. The scientific mass m_s consists of scientific equipments mass and penetrators mass. The available propellent mass m_p is calculated as follows,

$$m_p = m_i - m_d - m_s \tag{5.9}$$

The departure exceed velocity is up to 5 km/s. The minimum penetration relative velocity $\Delta v_{p_{min}}$ is 0.4 km/s. The maximum available thrust level $T_{c_{max}}$ is 0.3 N (i.e. $0 \leq T_c \leq 0.3$ N). The spacecraft has a constant specific impulse $I_{sp} = 3000$ s. The total mission period, measured from the Earth departure to the final asteroid maneuver, must not exceed 15 years.

The HGGA tool is used to obtain a preliminary impulsive trajectory. An initial set of more than 5000 asteroids is considered as a domain for the asteroid selection. To limit the design space, the asteroids set is limited based on their orbital parameters. First, the asteroids are arranged by their inclination. A smaller set of lower inclination asteroids are chosen. Then, the selected asteroid are arranged by their semi-major axis. A final set of the low energy asteroids are selected. The asteroids are coded as discrete design variables where each one is represented as a single integer number. The lower and upper bounds of the TOF of each leg are 100 and 1000 days, respectively. The accuracy of the TOF design variable is 5 days. The departure date accuracy is 10 days.

A population of 1000 individuals is used, for 1000 generations. The obtained solution is a trajectory completes 6 asteroids missions (i.e. 12 asteroid maneuvers are done). Each asteroid is visited twice, the first is a rendezvous and the second is a penetration. The total mission duration is 5390.1 days. The scientific mass is 246 kg (41 kg per asteroid). The departure date is 61034.71 MJD. The launch exceed velocity is 0.732 km/s. The visited asteroids are shown in Table 5.1. Figure 5.4 illustrate the impulsive solution obtained from the HGGA tool. Spacecraft distance from the Sun is drawn versus time for the suggested impulsive trajectory.

Table 5.1
Asteroids revisiting mission scenario obtained from HGGA tool

Asteroids	**Maneuver Date** (MJD)	**Maneuver Type**
(2009 BD)	61744.61	Rendezvous
(2006 RH120)	62024.02	Rendezvous
(2000 SG344)	62318.49	Rendezvous
(2008 UA202)	62574.41	Rendezvous
(2009 BD)	63212.03	Penetration
(2006 RH120)	63874.21	Penetration
(2008 UA202)	64590.72	Penetration
(2000 SG344)	65390.72	Penetration
(2008 JL24)	65627.92	Rendezvous
(2008 CM74)	65826.30	Rendezvous
(2008 JL24)	66124.67	Penetration
(2008 CM74)	66424.81	Penetration

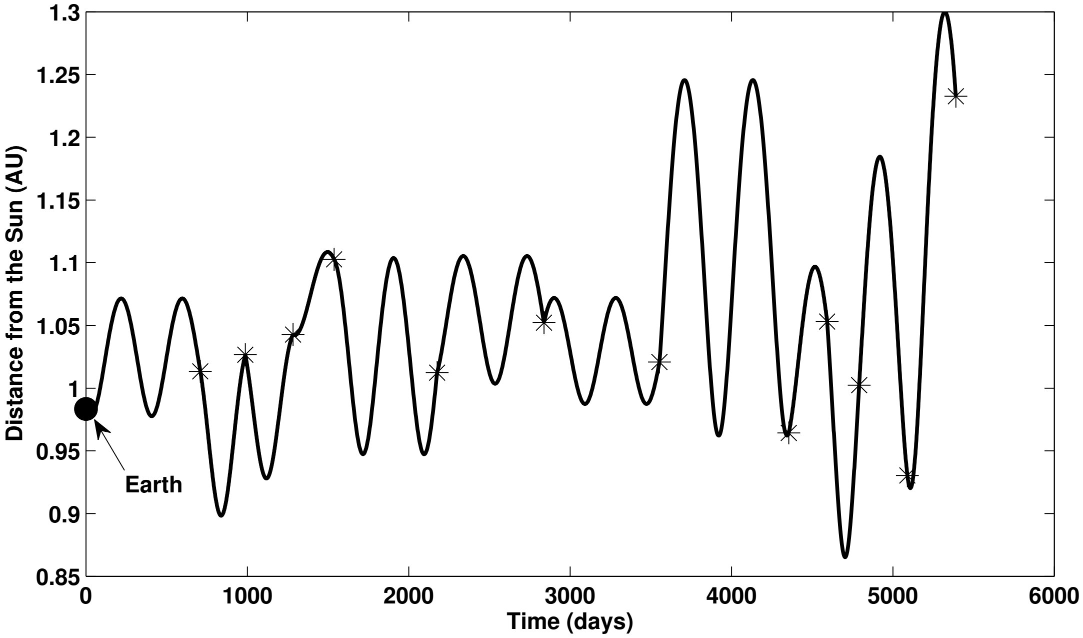

Figure 5.4: 12 visited asteroids and their distance from the Sun vs. time.

5.6 Summary

This chapter introduces an application for the GTOC problem. The asteroids revisiting mission is divided into four stages to obtain the optimal low thrust trajectory. The HGGA tool is used to generate an initial impulsive trajectory. The developed algorithm has the capability to find the number of visited asteroids, the maneuver type, and the date of maneuver. The solutions found using the developed HGGA tool is then used as an initial guess to obtain the low thrust trajectory of the GTOC problem.

Chapter 6

Optimal Earth Orbit Design for Regional Coverage Missions*

The problem of Earth orbit design is developed to select orbital elements which provide natural regional coverage for a set of ground sites, within a specified time frame (23). The spacecraft FOV is considered at regional coverage analysis. The J_2 effect is utilized to design a Sun-synchronous repeated ground track orbit that passes over the given set of ground sites. The selected design space consists of four independent variables: eccentricity e, inclination i_n, spacecraft's true anomaly above the first ground site ϑ_1, and the ground track repetition period in days, M_r. All given ground sites are checked whether is covered during the M_r days or not.

6.1 Sun-Synchronous Repeated Ground Track Orbits

For a given set of values for the four design variables (member in the design space), the orbit semi major axis, a, is constraint to be (51),

$$a = \left[-\frac{3R_E^2 J_2 \sqrt{\mu}}{2\left(1-e^2\right)^2 \dot{\Omega}} \cos i_n \right]^{2/7} \tag{6.1}$$

* The material contained in this chapter was previously published in the journal *Acta Astronautica, Elsevier* (23).

where $\dot{\Omega} = 1.991 \times 10^{-7}$ rad/sec is the desired rate of change of the orbital right ascension of the ascending node, in order to have a Sun-synchronous orbit. The constraint of having a repeated ground track orbit can be formulated as follows (38),

$$N_r|\Delta\phi| = 2\pi M_r \tag{6.2}$$

where N_r is the total number of successive orbit revolutions performed, M_r is the number of Earth revolutions (in days) before the ground track repeats itself, and $\Delta\phi$ is the total changes in longitude after one nodal period. $\Delta\phi$ is defined as (38),

$$\Delta\phi = \Delta\phi_1 + \Delta\phi_2 \ \text{ rad/orbit} \tag{6.3}$$

where $\Delta\phi_1$ and $\Delta\phi_2$ are the change in longitude due to the Earth's rotation and the regression of the line of node, respectively. They are computed according to a scheme described in Reference (38).

The above two constraints are implemented as follows. First, all the design variables are selected by the optimization tool. Then, the semi-major axis is computed using Equation (6.1) to generate a sun-synchronous orbit. This solution, however, does not yet satisfy the repeated ground track constraint, Equation (6.2). For any orbit, it is possible to slightly tune its inclination to obtain a repeated ground track orbit which is very close to the untuned orbit (30). Therefore, the inclination value is numerically tuned in order to generate a sun-synchronous repeated ground track orbit satisfying the problem constraints. Tuning the inclination value leads to a considerable change in the semi-major axis value according to Equation (6.1).

6.2 Intersection Locations Calculation

At this point, the unknown orbital elements are the argument of perigee ω and the right ascension of the ascending node Ω. A method of deriving ω and Ω from the known orbital elements , a, e, i_n, and ϑ_1, is stated in a previous development (42). All possible values of ω and Ω can be calculated numerically or analytically using the transformation theory from the inertial frame to the perifocal frame. The analytical method is used in this development to determine accurate values for the unknown orbital elements.

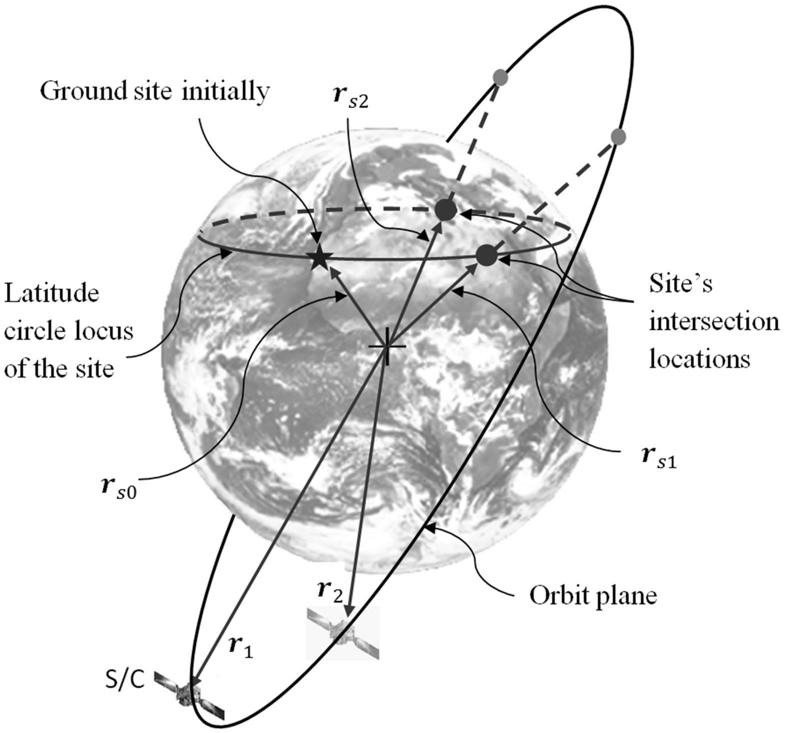

Figure 6.1: Two intersection locations of a fixed two-body orbit plane with the latitude locus circle of a ground site

The next step is to develop a method to check the regional coverage for each ground site. The longitude ϕ and the latitude λ for each ground site are given. At any point in time, a ground site is not, in general, in the spacecraft's orbit plane; and hence, at this time, it is not being visited. As Earth is spinning, each ground site moves on a circle normal to the Earth axis of rotation, which is the constant declination circle of that site, as shown in Figure 6.1. For each site, the latitude circle (same as the constant declination circle) intersects with the spacecraft orbit plane twice, in one orbital revolution. Due to Earth spinning and J_2 effect, the intersection location coordinates, for a given site, change over time, from one orbit revolution to the other. To check whether, or not, a ground site is visited, in the given time frame, all of its intersection points, with the candidate orbit plane, need to be accurately calculated, over the whole time frame. The site is considered visited if at least one of the intersection points is in the instantaneous coverage region of the spacecraft, at the intersection time.

To simplify the calculations of the intersection locations, we do it on two steps. First, the intersection locations are approximately calculated assuming a two-body motion. Then a numerical correction is applied to determine the correct intersection locations, for each site. In a two-body motion, the orbit plane is fixed over time. Therefore, all the orbital elements, but the true anomaly, are constant over time. With this assumption, we can use a closed form solution to find the intersection points (42). Two intersection locations are calculated, see Figure 6.1. Due to J_2 perturbation, the orbit plane changes over time. Hence, a correction is needed for the intersection locations. A numerical scheme is developed to achieve this correction. An iterative approach is implemented to calculate the time at which the site intersects with the rotating orbit plane. Figure 6.2 explains the idea of this scheme.

First, the spacecraft is propagated from $\mathbf{r}_1$ to a new position vector, $\mathbf{r}_2$. The propagation is for a guessed value for the time until intersection Δt, or for the true anomaly change, ϑ_{12}. The J_2 perturbation effect is taken into consideration in spacecraft propagation. The plane defined by the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ is an initial approximation for the orbit plane. This plane and the initial position vector of the site, $\mathbf{r}_{s0}$, are used to calculate the intersection position vectors, $\mathbf{r}_{s1}$ and $\mathbf{r}_{s2}$. This completes the first iteration. A new update for the time interval Δt between the initial site location $\mathbf{r}_{s0}$ and its intersection location $\mathbf{r}_{sj}$ with the orbit plane is calculated. Δt can be calculates as follows:

$$\Delta t = \frac{|\Delta\phi_{sj}|}{\omega_E} \tag{6.4}$$

where ω_E is the Earth spinning rate ($\omega_E = 7.2921 \times 10^{-5}$ rad/sec), and $\Delta\phi_{sj}$ is the change in site longitude due to Earth spinning, and is defined as:

$$\Delta\phi_{sj} = \phi_{sj} - \phi_{s0} \tag{6.5}$$

where ϕ_{s0} is the site's initial right ascension, and ϕ_{sj} is the intersection location right ascension. The latter can be calculated as follows:

$$\phi_{sj} = \tan^{-1}\left(\frac{\mathbf{r}_{sj}(2)}{\mathbf{r}_{sj}(1)}\right) \tag{6.6}$$

Using the updated value of Δt, the new orbit plane is calculated by propagating the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ by Δt. A more accurate approximation for the orbit plane is obtained. The J_2 effect is taken into consideration in propagation. The updated orbit plane and $\mathbf{r}_{s0}$ are used to calculate a new update for the intersection locations. The procedure is repeated until no improvements can be obtained. New intersection locations are computed, for each day in the given time frame M_r. So, there are $2M_r$ intersection points between the orbit

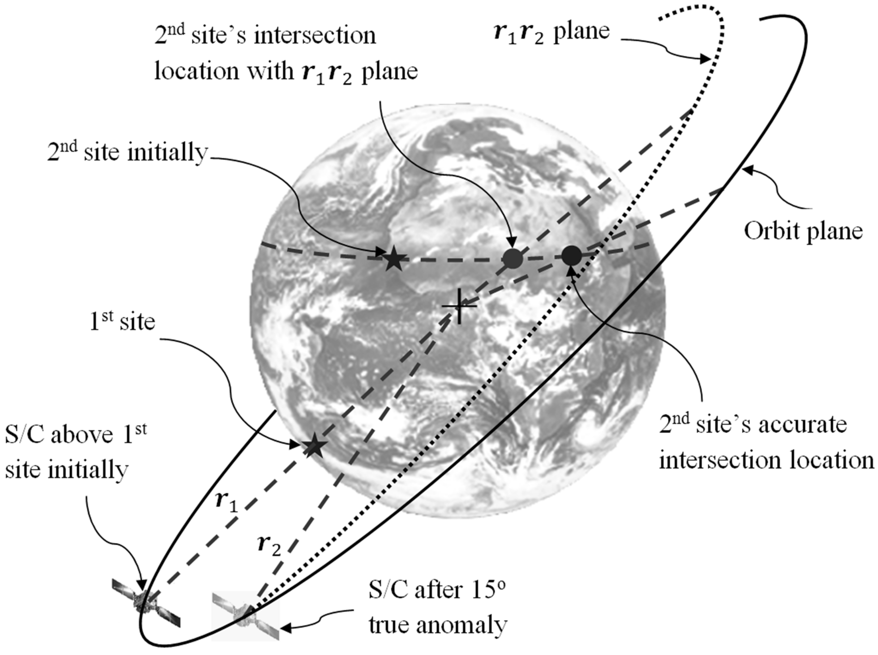
2nd site's intersection location with $r_1 r_2$ plane
$r_1 r_2$ plane
2nd site initially
Orbit plane
1st site
2nd site's accurate intersection location
S/C above 1st site initially
r_1
r_2
S/C after 15° true anomaly

Figure 6.2: Ground site's intersection locations with spacecraft perturbed trajectory

plane and each site, within M_r days time frame. Recall that at each design point, we have two possible orbit planes. At this step, we know the times at which the sites intersect with the orbit plane. The next step is to check whether each sites will be covered by the spacecraft or not, at the times of intersections. If all sites are covered, then the candidate orbit is a solution orbit. The next section briefs how to calculate the coverage for a site.

6.3 Coverage with Sensor's FOV

The field of view FOV of the spacecraft is defined as the angular distance viewed by the instrument installed on the spacecraft. Figure 6.3 shows the definitions and angular relationships between the spacecraft, Earth's center, and coverage zone. The angular radius of the Earth ρ and the maximum Earth central angle λ_o can be calculated from the geometric relation:

$$\sin\rho = \cos\lambda_o = \frac{R_E}{R_E + H} \tag{6.7}$$

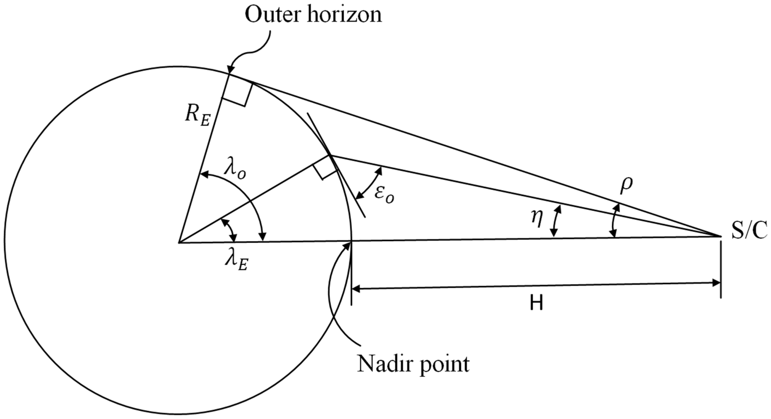

Figure 6.3: Definition of angular relationships between the spacecraft, Earth's center, and coverage zone

The nadir angle η (also called half of the angular FOV) is measured at the spacecraft from the nadir point to the end of coverage zone. The spacecraft elevation angle ε_o is the angle measured at coverage ending zone between the spacecraft and the local horizontal. Given η, ε_o can be computed as (56),

$$\cos \varepsilon_o = \frac{\sin \eta}{\sin \rho} \tag{6.8}$$

The Earth central angle λ_E is measured at the center of the Earth from the nadir point to the end of coverage zone. λ_E can be computed as:

$$\lambda_E = \frac{\pi}{2} - \eta - \varepsilon_o \tag{6.9}$$

Let $\mathbf{r}_{s/c}$ and $\mathbf{r}_{sj}$ be the spacecraft and site position vector, respectively, at the time we check the coverage. The angle θ_{cov} is the angle between these two vectors, and can be computed as follows:

$$\theta_{cov} = \frac{\mathbf{r}_{s/c} \cdot \mathbf{r}_{sj}}{r_{s/c} r_{sj}} \tag{6.10}$$

If the angle θ_{cov}, for a certain site, is less than or equal to the Earth central angle λ_E, then this site is covered by the spacecraft, at this time.

6.4 Optimization

This section describes how we optimize the selection of the independent design variables. The purpose is to design a Sun-synchronous repeated ground track orbit that covers as many sites as possible, from a given set of ground sites, in minimum mission time. The parameters to be optimized are: eccentricity e, inclination i_n, spacecraft's true anomaly above the first ground site ϑ_1, and the ground track repetition period in days, M_r. The fitness F_i (the cost function to be maximized) is calculated as follows:

$$F_i = \frac{1.5 N_c(e, i_n, \vartheta_1, M_r)}{M_r^{0.1}} \tag{6.11}$$

GA is implemented to find a highly fit solution for the problem. Each member, design point, in the population represents four design variables. Three of them are the orbital elements that are coded as CDVs. These variables are e, i_n, and ϑ_1. The mission time frame, M_r, is coded as a DDV. The design variables coding is performed as described in Section 3.1. The inclination should be more than 90° to provide the desired regression of nodes of a Sun-synchronous orbit. Also, a limitation on the eccentricity, to be less than 0.6, is applied to guarantee certain minimum altitude for the spacecraft. The values of the four design variables are used to evaluate the unknown orbital elements and check the coverage of each ground site. This process is carried out as described in the previous sections. Then, the fitness function F_i, for each member i in the population, is calculated using Equation (6.11). The fittest members in the population are selected as parents for the next generation. Genetic algorithms operations are then used to generate a new generation. This process is repeated for a predefined number of generations. Finally, the fittest members in the last generation are selected. The most fit design point is the member which represents a Sun-synchronous repeated ground track orbit that optimizes the objective function.

6.5 Numerical Results

As a challenging numerical example of designing J_2 perturbed orbits, a set of 20 ground sites is randomly selected. The field of view (FOV) is selected to be 10^o. The maximum possible value of M_r is selected to be 32 days. The population size is selected to be 100, and the number of generations is 200. The probability of crossover is selected to be 0.9, while the mutation probability is in the range of 0.01 and 0.08. The most fit 10% of the members in each generation are copied automatically into the next generation without performing any genetic operations. Equation (6.11) is used as the fitness function to be maximized. The fittest orbit obtained by the GA is an orbit that covered the 20 sites in 25 days. This

Table 6.1
The orbital elements of a sample of the solutions, for the 20 sites problem.

Orbit Parameters	Fittest Orbit	Alternative Orbits		
Semimajor axis a (km)	10031.187	10025.712	9935.833	11081.923
Eccentricity e	0.1935	0.1935	0.1419	0.1935
Inclination i_n (deg)	116.55	116.50	116.62	129.31
Right ascension of ascending node Ω(deg)	263.41	348.60	263.45	344.55
Perigee argument ω (deg)	306.77	79.254	306.84	61.611
True anomaly of the first site ϑ_1 (deg)	141.17	146.82	141.17	141.17
No. of covered sites	20/20	12/20	11/20	9/20
Ground track repetition, **period**, M_r (days)	25	17	17	9

resulted in 12 hr of computing time. The orbital elements of the fittest solution are listed in Table 6.1. The developed algorithm can be used to provide many other alternative orbits which cover less number of ground sites in less repetition period. Table 6.1 shows the orbital elements of a sample of these alternative orbits. To generate these results, several trials have been attempted with different selections for the genetic operations parameters.

6.6 Summary

In this chapter, natural Earth orbits is designed for zonal coverage missions. The problem is formulated to maximize the number of visited ground sites and to minimize the mission time frame. J_2 perturbation effect is implemented to design sun-synchronous repeated ground track orbits. The field of view of the sensor on board the spacecraft is considered in the coverage analysis. GA is utilized as an optimization technique to handle this problem. The developed algorithm has the capability to solve a challenging case study with twenty ground sites.

Chapter 7

Repeated Shadow Track Orbits for Space-SunSetter Missions[*]

This chapter introduces a new set of orbits, the "Repeated Shadow Track Orbits" (30). In these orbits, the shadow of a spacecraft on the Earth visits the same locations periodically every desired number of days. The J_2 perturbation is utilized to synchronize the spacecraft shadow motion with both the Earth rotational motion and the Earth-Sun vector rotation. The well known repeated ground track orbits have ground tracks that repeat every given number of days. The orbital elements may be selected such that the total change in the longitude of the nadir point, after an integer number of nodal periods, is an integer multiple of one complete Earth rotation about its axis (38). The fact that we are using the nodal period results in no change in the nadir point latitude after any integer number of complete nodal periods. In the case of a repeated shadow track orbit, the Shadow-Nodal period is introduced. Shadow-Nodal period is the time it takes the shadow of a spacecraft to come back to the same latitude, after orbiting the Earth once. In repeated shadow track orbits (RSTO), the orbital elements are selected such that the total change in the longitude of the shadow point, after an integer number of Shadow-Nodal periods, is an integer multiple of one complete Earth rotation about its axis. The orbit parameters selection of an RSTO is affected by the Earth's rotation rate around its axis as well as its rotation rate around the Sun.

[*] The material contained in this chapter was previously published in the *International Journal of Aerospace Engineering* (30).

7.1 Shadow Location Calculations

The spacecraft should be located on the line from the Earth to the Sun to obtain a shadow point on the Earth surface. The angle between the spacecraft position vector and the Sun position vector is defined as the shadow angle θ_{sh}, measured in the Earth Centered Inertial frame (ECI). Figure 7.1 illustrates the geometry for shadow location calculations. The shadow position vector $\mathbf{r}_{sh}$ can be calculated at any particular time by defining the spacecraft position vector $\mathbf{r}_s$, and the Sun position vector $\mathbf{r}_{sun}$. The magnitude of the shadow position vector is the Earth radius R_E.

$$\|\mathbf{r}_{sh}\| = R_E \tag{7.1}$$

From Figure 7.1, the shadow angle θ_{sh} can be calculated as follows:

$$\cos\theta_{sh} = \hat{r}_s \cdot \hat{r}_{sun} \tag{7.2}$$

where $\hat{r}_s$ and $\hat{r}_{sun}$ are the unit vectors in direction of spacecraft and the Sun, respectively. For the oblique triangle formed by $\mathbf{r}_s$ and $\mathbf{r}_{sh}$,

$$R_E^2 = z^2 + r_s^2 - 2zr_s\cos\theta_{sh} \tag{7.3}$$

Therefore,

$$z^2 - (2r_s\cos\theta_{sh})z + (r_s^2 - R_E^2) = 0 \tag{7.4}$$

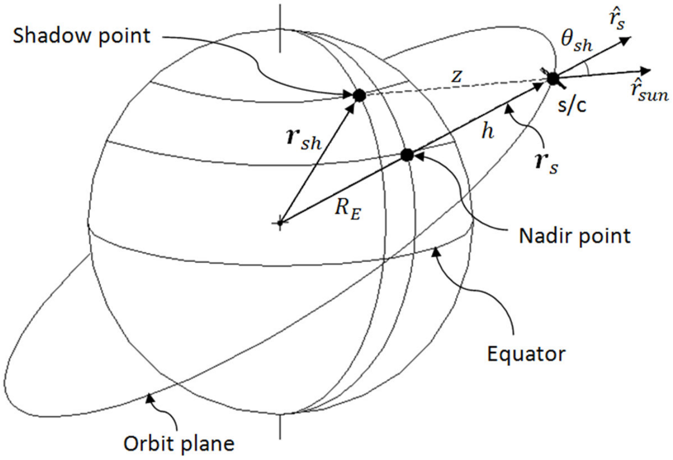

Figure 7.1: Shadow position vector, $\mathbf{r}_{sh}$, and Shadow angle θ_{sh}

$$z = r_s \cos\theta_{sh} \pm \sqrt{R_E^2 - r_s^2 \sin^2\theta_{sh}} \tag{7.5}$$

where R_E is the Earth's radius, and z is the distance between the spacecraft and its shadow point on the Earth's surface. The distance z has two values; the smaller positive real value is chosen to calculate the shadow position vector:

$$\mathbf{r}_{sh} = \mathbf{r}_s - z\hat{r}_{sun} \tag{7.6}$$

To get a shadow on the Earth's surface, the shadow angle should not exceed θ_{lim}. The angle θ_{lim} is calculated as follows:

$$\theta_{lim} = sin^{-1}\left(\frac{R_E}{h + R_E}\right) \tag{7.7}$$

where h is the spacecraft altitude measured from the Earth surface, and θ_{lim} is in the first quadrant.

7.2 Repeated-Shadow Sun-synchronous Orbits

The shadow of a spacecraft in a repeated shadow track orbit repeats its track every certain period of time. If this period of time is an integer number of days, then the orbit may be called a repeated-shadow Sun-synchronous orbit. Therefore, after an integer number of successive shadow revolutions, and an integer number of days, the spacecraft shadow repeats its shadow track. A repeated-shadow Sun-synchronous orbit is favored for those missions where the shadow is required to revisit ground regions at the same daylight conditions. To derive the necessary conditions for a repeated-shadow Sun-synchronous orbit, we need to take into consideration the Earth rotation about its axis, the orbit parameters perturbations due to J_2 effect, and the Earth rotation around the Sun (38).

The Earth rotates through one revolution in its sidereal period of τ_E , where $\tau_E = 86164.1$ sec. The effect of the Earth's rotation around its axis is a change in shadow point longitude $\Delta\varphi_1$ after one orbital revolution. Whereas there is no change in shadow point latitude, i.e. $\Delta\lambda_1 = 0$. The second effect on RSTO is the orbit perturbations (38). After one spacecraft orbital period T, and due to only J_2 perturbation, the shadow track will be changed by $\Delta\varphi_2$ in longitude. The third effect on RSTO is the Earth's rotation around the Sun (51). The Earth completes one revolution around the Sun once every year. The Earth rotates around the Sun in the ecliptic plane while Earth's equatorial plane is inclined about 23.5° to the ecliptic. This phenomenon explains the change in both latitude $\Delta\lambda_3$ and longitude $\Delta\varphi_3$ of a spacecraft's shadow point on earth surface after one nodal period. More details regarding this configuration are explained in Reference (30).

The total changes in shadow point latitude $\Delta\lambda_{nodal}$ and longitude $\Delta\varphi_{nodal}$ after one spacecraft nodal period τ are

$$\Delta\varphi_{nodal} = \Delta\varphi_{1\,nodal} + \Delta\varphi_{2\,nodal} + \Delta\varphi_{3\,nodal} \text{ rad/orbit} \tag{7.8}$$

$$\Delta\lambda_{nodal} = \Delta\lambda_{3\,nodal} \text{ rad/orbit} \tag{7.9}$$

To select a repeated shadow track orbit, the shadow track should revisit the same ground locations, latitude and longitude, after an integer number of shadow revolutions within a certain period of time (integer number of days for Sun-synchronous orbits). Thus, it is required that some integral number of orbits later the accumulated value of the total change in shadow longitude will equal 2π, and the total change in shadow latitude will equal zero (38). Therefore, the total change in shadow longitude $\sum_{i=1}^{n_{sh}} |\Delta\varphi_{shadow}|$ and the total change in shadow latitude $\sum_{i=1}^{n_{sh}} |\Delta\lambda_{shadow}|$ can be formulated as following,

$$\sum_{i=1}^{n_{sh}} |\Delta\varphi_{shadow}| = 2\pi N_m \tag{7.10}$$

$$\sum_{i=1}^{n_{sh}} |\Delta\lambda_{shadow}| = 0 \tag{7.11}$$

where n_{sh} is the total number of successive shadow revolutions performed and N_m is the number of Earth revolutions (equivalent to days) before an identical shadow track occurs.

After completing one shadow revolution, the shadow track changes only by $\Delta\varphi_{sh}$ in longitude, while the change in shadow latitude $\Delta\lambda_{sh}$ is equal to zero. The period needed to complete one shadow revolution, by keeping $\Delta\lambda_{sh} = 0$, is the shadow nodal period τ_{sh}. It can be calculated from the following formula,

$$\tau_{sh} = \frac{2\pi}{\omega_{sh}} \tag{7.12}$$

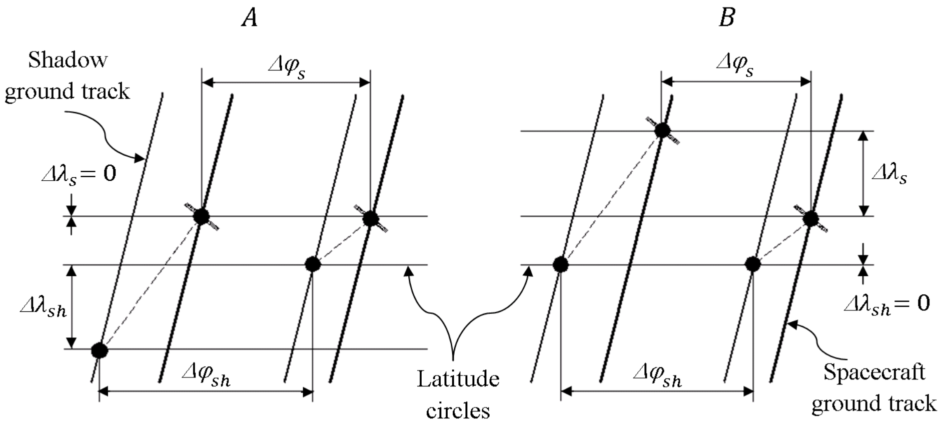

Figure 7.2: Latitude and longitude changes for shadow and spacecraft ground tracks after: (A) spacecraft nodal period τ, (B) shadow nodal period τ_{sh}

where ω_{sh} is the rate of change of ground shadow point position measured in the shadow track on Earth surface. ω_{sh} depends on two main parameters. The first is the rate of change of the spacecraft position along the ground shadow track orbit. The second is the rate of change of the Earth-Sun vector in the same orbit. Figure 7.2 shows latitude and longitude changes for both shadow and spacecraft ground track after shadow nodal period τ_{sh} and spacecraft nodal period τ. Shadow point changes in both latitude and longitude with different rate from the sub-spacecraft point "nadir". This phenomenon is a direct result of variation of Earth-Sun vector with time.

A numerical approach is implemented to calculate the shadow nodal period τ_{sh} of a spacecraft has a nodal period τ. As illustrated in Figure 7.2, the change in shadow latitude of a spacecraft propagated for τ_{sh} is equal zero. Hence, propagating a spacecraft for any other period results in a non-zero change in shadow point latitude. This fact is used to calculate an accurate value for a spacecraft shadow nodal period at a certain day time. The following formula is used to get an accurate value for τ_{sh},

$$T = T\left[1 - \frac{\Delta\lambda_{sh}}{2\pi}\right] \tag{7.13}$$

First, the change in shadow latitude $\Delta\lambda_{sh}$ is calculated after propagating the spacecraft for a nodal period τ. Then, the spacecraft nodal period is substituted in the right hand side of Equation (7.13) with its corresponding change in shadow latitude. The calculated value of T in the left hand side can be considered as an initial value of shadow nodal period τ_{sh}. This initial value is used to propagate the spacecraft and calculate its corresponding

$\Delta\lambda_{sh}$. Once again, Equation (7.13) is used to generate a correlated value of the shadow nodal period. Finally, an accurate value of τ_{sh} can be calculated after a limited number of iterations. At this point, the total change in shadow location after τ_{sh} is in shadow longitude $\Delta\varphi_{sh}$, while $\Delta\lambda_{sh}$ equals to zero.

From the previous discussion, we realize that the shadow nodal period τ_{sh} depends on the orbital elements and on the Sun's position, which is a function of time. It can be concluded then that the shadow nodal period of a spacecraft orbiting the Earth is also a function of time. Hence, the change in shadow point longitude after propagating the spacecraft for τ_{sh} is also a function of time.

7.3 Optimization

For the purpose of designing an RSTO that has its shadow repeats over a certain ground site with a maximum duration for the shadow per day over this site and a minimum number of revolutions per day, an optimization problem is formulated. The parameters to be optimized are the spacecraft orbital elements. These parameters are subjected to constraints to provide the repeated shadow track conditions. The fitness F_i (the cost function to be maximized) at a design point is defined as follows:

$$F_i = f_1 \frac{T_{tsh}}{N_m} - f_2 \frac{n_{sh}}{N_m} \tag{7.14}$$

where f_1 & f_2 are the fitness weight parameters, T_{tsh} is the target shadow period over a specific site, n_{sh} is the number of successive shadow revolutions, and N_m is the number of days before an identical shadow track will occur. The previous expression is valid only for the design points which provide the repeated shadow track conditions. Otherwise, the fitness F_i for any non-RSTO design point is chosen to be a small negative value less than the minimum value of the fitness function for any RSTO design point.

Ga is used to obtain the optimal solution of this problem. Each member in the GA population represents six design variables. Five are orbital elements that are coded as CDVs. These variables are eccentricity e, inclination i_n, right ascension of ascending node Ω, argument of perigee ω, and true anomaly ϑ. The desired number of days before an identical shadow track will occur, N_m, is coded as a DDV. The design variables coding is applied as described in Section 3.1. The inclination should be more than 90° to provide the desired regression of node of a Sun-synchronous orbit. Also, a limitation on the eccentricity to be less than 0.6 is applied to guarantee certain minimum altitude for the spacecraft. The design variables values are used to evaluate the fitness function F_i for each design point separately according to the following steps. First, the semi-major axis is calculated to provide the condition of a Sun synchronous orbit. The perigee altitude is determined to check the feasibility of the orbit. The semi-major axis a can be calculated for each design point from Equation 6.1.

Now, the spacecraft location for each design point is known from the 6 orbital elements. Hence, shadow calculations, discussed in Section 7.1, is used to decide whether that design point has a shadow on the Earth's surface or not. For any non-shadow members, the fitness is chosen to be a small negative value. The ground shadow location is determined for those members satisfying ground shadow at a predefined initial time. Therefore, the shadow nodal period τ_{sh} can be calculated according to the numerical technique described in Section 7.2. Then, the shadow track is propagated by τ_{sh} over a limited number of revolutions n_{sh} until the total change in shadow longitude $\sum_{i=1}^{n_{sh}} |\Delta\varphi_{shadow}|$ accumulates $2\pi N_m$ or more. Using τ_{sh} in shadow propagation guarantees total change in shadow latitude $\sum_{i=1}^{n_{sh}} |\Delta\lambda_{shadow}|$ equal to zero. The total number of shadow revolutions can be considered as the integer number of orbits required for shadow repetition in N_m days. The summation of shadow nodal period $\sum_{i=1}^{n_{sh}} \tau_{sh}$ is determined, which is equal to or greater than N_m days.

Not all design point in the population satisfy the repeated shadow track conditions. Some members do not have a shadow track on Earth surface. The rest of the design points, which provide ground shadow track, have $\sum_{i=1}^{n_{sh}} |\Delta\varphi_{shadow}| > 2\pi N_m$, or $\sum_{i=1}^{n_{sh}} \tau_{sh} > N_m$. In this case, an equivalent design point is calculated to check whether this orbit could be modified to an RSTO or not. This is orbital elements tuning. It is conducted by changing (tuning) only one orbital element to satisfy the repeated shadow track conditions. The inclination is selected to be the tuning element. Changing the inclination value leads to a change in the semi-major axis to keep the Sun synchronous orbit condition. The inclination is tuned to guarantee the total change in shadow longitude accumulates $2\pi N_m$ within an acceptable error tolerance. The orbital parameters tuning is not enough to decide that the equivalent design point represent an RSTO. The summation of shadow nodal period $\sum_{i=1}^{n_{sh}} \tau_{sh}$ for the modified design point must equal to the integer number N_m. satisfying these conditions guarantees that the design point in the population corresponds to a repeated shadow track orbit.

The target shadow period T_{tsh} is required to calculate the fitness of the RSTOs design points. To calculate T_{tsh}, the spacecraft's shadow is propagated for the whole N_m days required for shadow repetition. Then, the target shadow period is measured as the total time of shadow existence over a target site in N_m days. To get an accurate value of T_{tsh}, the orbit propagation should be performed using a small time step, about 1 sec. Given the problem in question, this can be quite time consuming. To minimize the run time, a varying time step is used during shadow propagation. For the shadow points which are close enough to the target site, a small time step is used in propagation. When the shadow is not a concern, a larger time step is used. Finally, by calculating T_{tsh}, N_m, and n_{sh}, the fitness F_i can be determined for each design point. The fittest members in the initial population are selected as parents for the next generation. Genetic algorithms operations are then used to generate a new generation which contains better design points. This process is repeated for a predefined number of generations. Finally, the fittest members in the last generation are selected. The most fit design point is the member which represents a repeated shadow track orbit passes over a particular target site with minimum number of revolutions and maximum target shadow period at a given time.

7.4 Numerical Results

GA are used to select the best RSTO appropriate for a specific ground site at a certain date. The most fit orbital elements should provide a shadow repetition with a minimum number of shadow revolutions during an integer number of days and a maximum shadow duration over the given ground site. A numerical example is performed to generate the best orbital elements of an RSTO starting at the first day of July 2008. The ground site has longitude and latitude of 51.3484° and 25.1528° respectively. The site shadow duration T_{tsh} is calculated based on the assumption that the site size is 10 km radius from the site center. The shadow duration is the time during which the shadow is within this radius.

The population size is selected to be 100, and the number of generation is 200. The probability of crossover is considered 0.9, while the mutation probability is taken between 0.01 and 0.08. 10% of the most fit members in each generation are copied automatically into the next generation without performing any genetic operations. The values of the weight parameters f_1 and f_2 are selected to be 2 and 1, respectively. The fittest orbital obtained by genetic algorithms represents an RSTO that passes over the given site. This

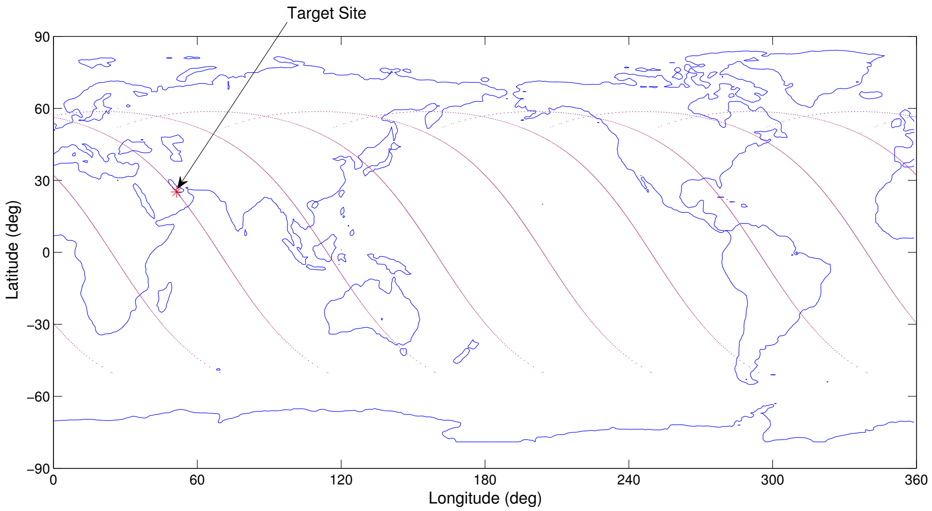

Figure 7.3: The ground shadow track for the repeated shadow track orbit obtained by genetic algorithm.

orbit completes 8 successive shadow revolutions in one day to provide shadow repetition over the desired site. The orbital elements of the obtained solution are: eccentricity 0.2505, semi-major axis 10560.2095 km, inclination 120.4955°, argument of perigee 0.0879°, ascending node 113.2609°, and true anomaly 22.8544°. The total shadow duration over the Earth surface is 14560 seconds per day. While the site shadow duration is six seconds per day. Figure 7.3 shows the ground shadow track for the RSTO obtained by genetic algorithm.

7.5 Summary

In this chapter, a new Earth orbit category is presented to investigate the Space-SunSetter mission. The repeated shadow track orbit is formulated and proved. Orbital elements constraints are derived to obtain a repeated shadow track orbit. J_2 perturbation is utilized to guarantee the repetition of the shadow track on Earth surface. The term of shadow nodal period is introduced to investigate the new orbit characteristics. GA is used as an optimization technique to maximum time duration over a specified ground site.

Chapter 8

Conclusions

The problem of optimizing space trajectories is presented in this dissertation. Genetic algorithm is used as an optimization technique to obtain the optimal space trajectories. Two categories of space trajectories are addressed, the interplanetary space trajectories and the Earth orbiting trajectories. GA is utilized to investigate complex space trajectories optimization problems with variable-size design space. The problem of optimal design of a multi-gravity-assist space trajectory, with free number of deep space maneuvers is investigated. The motivation of this study is to develop an optimization algorithm that can, without a priori knowledge, compute the number of swing-bys and the planets to swing by, in addition to the rest of the classical MGADSM design variables. This is a complex problem characterized by the following: first, some of the design variables are discrete, second, the number of design variables is solution-dependent, and finally the number of design variables becomes rather high in complex missions. Solution-dependent design variables mean that different solutions have different number of design variables. To handle global optimization problems where the number of design variables varies from one solution to another, two novel genetic-based techniques are developed: hidden genes genetic algorithm (HGGA) and dynamic-size multiple population genetic algorithm (DSMPGA). Both techniques have the capability to handle mixed (discrete and continuous) design variable optimization problems, with variable size design space.

The interplanetary trajectory optimization problem is addressed in Chapter 3 using a novel hidden genes optimization technique. The concept of hidden genes genetic optimization proved to be capable of finding, without a priori knowledge, the number of swing-bys, the planets to swing by (optimal planet sequence), and the number of deep space maneuvers (DSMs) in each leg, as well as their components and directions, in addition to the rest of the design variables in the interplanetary trajectory optimization problem. A fixed chromosome length is assumed for all chromosomes in the population. Part of the chromosome is effective in fitness function evaluations, while the other part (hidden genes part) is ineffective. Yet, the hidden genes take part in the genetic operations. In some problems, a niching technique is needed to increase the diversity in the population, and hence increases the speed of convergence. In the three case studies presented in Chapter 3, the ITO-HGGA found the known optimal solutions with improvements in some cases. To avoid long computational time, in some complex problems, a two-phase algorithm was implemented. The first phase finds only the mission scenario (planet sequence), while the second algorithm determines the DSMs structure in the trajectory.

The interplanetary trajectory optimization problem is addressed in Chapter 4 using a new dynamic-size multiple population genetic algorithm. A genetic algorithm is applied to a calculated number of sub-populations in parallel. Each sub-population represents a certain size of the design space (certain number of design variables). The initial size of each sub-population is proportional to the number of design variables in the sub-population. The size of a sub-population then increases or decreases in subsequent stages depending on the fitness of the members of the sub-population. The DSMPGA method proved to be capable of finding, without a priori knowledge, the number of swing-bys, the planets to swing by (optimal planet sequence), and the number of deep space maneuvers (DSMs) in each leg, as well as their components and directions, in addition to the rest of the design variables in the interplanetary trajectory optimization problem. In the three case studies presented in Chapter 4, the DSMPGA found the known optimal solutions with improvements in some cases.

Chapter 5 introduces an application for the hidden genes concept. The developed hidden genes genetic algorithm tool is used to solve the GTOC problem. The asteroids revisiting mission is investigated to obtain the optimal trajectory with maximum number of visited asteroids. The GTOC problem is a good application for the concept of hidden genes. The developed tool is used to compute an impulsive trajectory, which could be used as an initial guess to obtain the continuous low thrust trajectory. The HGGA tool has the capability to find the number of visited asteroids, the maneuver type, and the date of maneuver.

In Chapter 6, the problem of initial orbit design for regional coverage, using natural (no thrust) orbits, is investigated. A novel problem formulation is developed to design an orbit that covers as many ground sites as possible, from given set, and also minimizes the mission time frame. The J_2 perturbation effect is implemented as a constraint to guarantee Sun-synchronous repeated ground track orbit solutions. The spacecraft's Field of View is an input parameter to the developed algorithm. The developed method demonstrated success in finding fit solutions for challenging case studies. One useful feature about implementing the genetic algorithms in optimization is that it provides several solutions to the problem, presenting different interesting features.

The concept of repeated shadow track orbit is introduced in Chapter 7. Constraints on the orbital elements are derived to obtain a repeated shadow track orbit. For a two-body motion perturbed only by the J_2, the orbital elements need to be updated continuously to maintain the repetition of the shadow track on ground. An optimization tool has been developed using a genetic algorithm approach to obtain the orbit with maximum time duration over a given ground site. Results show that for a natural orbit, *i.e.* without a control, the maximum duration time for a shadow over a point on ground will be in the order of few seconds in one orbit revolution.

Bibliography

[1] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman, Inc, 1989.

[2] Ely, T. A., Crossley, W. A., and Williams, E. A., “Satellite Constellation Design for Zonal Coverage Using Genetic Algorithms,” *Journal of the Astronautical Sciences*, Vol. 47, No. 3, Jul-Dec 1999, pp. 207–228.

[3] Kim, Y. H. and Spencer, D. B., “Optimal Spacecraft Rendezvous Using Genetic Algorithms,” *Journal of Spacecraft and Rockets*, Vol. 39, No. 9, Nov. 2002, pp. 859–865, doi:10.2514/2.3908.

[4] Kim, R., lung, O., and Bang, H., “A Computational Approach to Reduce the Revisit Time Using a Genetic Algorithm,” *International Conference on Control, Automation and Systems*, Oct. 2007, pp. 184–189.

[5] Abdelkhalik, O. and Mortari, D., “Orbit Design for Ground Surveillance Using Genetic Algorithms,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, September–October 2006, pp. 1231–1235, doi: 10.2514/1.16722.

[6] Rauwolf, G. and Coverstone-Carroll, V., “Near-Optimal Low-Thrust Orbit Transfer Generated by a Genetic Algorithms,” *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, Nov.-Dec. 1996, pp. 859–862, doi:10.2514/3.26850.

[7] Gad, A. and Abdelkhalik, O., “Hidden Genes Genetic Algorithm for Multi-Gravity-Assist Trajectories Optimization,” *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, August–Septemper 2011.

[8] Wall, B. J. and Conway, B. A., “Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics,” *Journal of Global Optimization*, Vol. 44, No. 4, August 2009, pp. 493 – 508, doi: 10.1007/s10898-008-9352-4.

[9] Vasile, M., Minisci, E., and Locatelli, M., “Analysis of Some Global Optimization Algorithms for Space Trajectory Design,” *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, Mar.-Apr. 2010, pp. 334–344, doi: 10.2514/1.45742.

[10] Abdelkhalik, O. and Gad, A., "Dynamic-Size Multiple Population Genetic Algorithm for MGADSM Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, 2011, Accepted.

[11] Battin, R. H., *Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, New York, 1987.

[12] Navagh, J., "Optimizing Interplanetary Trajectories with Deep Space Maneuvers," *NASA Contractor Report 4546, NASA Langley Research Center*, Hampton, VA, 1993.

[13] Abilleira, F., "Broken-Plane Maneuver Applications for Earth to Mars Trajectories," *20th International Symposium on Space Flight Dynamics*, September 2007.

[14] Izzo, D., Becerra, V., Myatt, D., Nasuto, S., and Bishop, J., "Search Space Pruning and Global Optimisation of Multiple Gravity Assist Spacecraft Trajectories," *Journal of Global Optimisation*, Vol. 38, No. 2, June 2007, pp. 283–296, doi:10.1007/s10898-006-9106-0.

[15] Olympio, J. T. and Marmorat, J.-P., "Global Trajectory Optimisation: Can we prune the solution space when considering Deep Space Maneuvers? Final Report," *"European Space Agency"*, Sep. 2007.

[16] Olympio, J. T. and Izzo, D., "Designing Optimal Multi-Gravity-Assist Trajectories with Free Number of Impulses," *21st International Symposium on Space Flight Dyncamics*, Toulouse, France, Sept 2009.

[17] Vasile, M. and Pascale, P. D., "Preliminary Design of Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 4, JulyAugust 2006, pp. 794–805, doi:10.2514/1.17413.

[18] Olds, A., Kluever, C., and Cupples, M., "Interplanetary Mission Design Using Differential Evolution," *Journal of Spacecraft and Rockets*, Vol. 44, No. 5, 2007, pp. 1060–1070, doi:10.2514/1.27242.

[19] Abdelkhalik, O. and Mortari, D., "N-Impulse Orbit Transfer Using Genetic Algorithms," *Journal of Spacecraft and Rocket*, Vol. 44, No. 2, Mar.-Apr. 2007, pp. 456–459.

[20] Gad, A., Addanki, N., and Abdelkhalik, O., "N-Impulses Interplanetary Orbit Transfer Using Genetic Algorithm with Application to Mars Mission," *20th AASAIAA Space Flight Mechanics Meeting*, No. AAS-10-167, San Diego, CA, Feb 2010.

[21] Zhu, K., Li, J., and Baoyin, H., "Trajectory Optimization of the Exploration of Asteroids Using Swarm Intelligent Algorithms," *Tsinghua Science and Technology*, Vol. 14, No. 2, 2009, pp. 7–11, 10.1016/S1007-0214(10)70022-5.

[22] Gantovnik, V. B., Gurdal, Z., Watson, L. T., and Anderson-Cook, C. M., "Genetic Algorithm for Mixed Integer Nonlinear Programming Problems Using Separate Constraint Approximations," *AIAA journal*, Vol. 43, No. 8, 2005, pp. 1844–1849, doi: 10.2514/1.4191.

[23] Abdelkhalik, O. and Gad, A., "Optimization of Space Orbits Design for Earth Orbiting Missions," *Acta Astronautica, Elsevier*, Vol. 68, April-May 2011, pp. 1307–1317, doi:10.1016/j.actaastro.2010.09.029.

[24] Dasgupta, D. and McGregor, D., "SGA : A Structured Genetic Algorithm," *Parallel Problem Solving from Nature*, September 1992, pp. 145–154.

[25] Kim, I. and deWeck., O. L., "Variable Chromosome Length Genetic Algorithm for Progressive Refinement in Topology Optimization," *Structural and Multidisciplinary Optimization*, Vol. 29, No. 6, 2005, pp. 445–456, 10.1007/s00158-004-0498-5.

[26] Ryoo, J. and Hajela, P., "Handling Variable String Lengths in GA Based Structural Topology Optimization," *Structural and Multidisciplinary Optimization*, Vol. 26, No. 5, 2004, pp. 318–325, doi:10.1007/s00158-003-0307-6.

[27] Brie, A. H. and Morignot, P., "Genetic Planning Using Variable Length Chromosomes," *ICAPS'05*, 2005, pp. 320–329.

[28] Ahn, C. W. and Ramakrishna, R. S., "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 6, 2002, pp. 566–579, doi:10.1109/TEVC.2002.804323.

[29] Katari, V., Ch, S., Satapathy, R., Murthy, J., and Reddy, P., "Hybridized Improved Genetic Algorithm with Variable Length Chromosome for Image Clustering Abstract," *IJCSNS International Journal of Computer Science and Network Security*, Vol. 7, No. 11, 2007, pp. 121–131.

[30] Gad, A. and Abdelkhalik, O., "Repeated Shadow Track Orbits for Space-SunSetter Missions," *International Journal of Aerospace Engineering*, Vol. 2009, No. 561495, 2009, pp. 13, 10.1155/2009/561495.

[31] Schaub, H.and Junkins, J., *Analytical Mechanics Of Space systems*, AIAA, 2003.

[32] Wie, B., *Space Vehicle Dynamics and Control*, AIAA Educational Series, Reston, VA, 1998.

[33] Wertz, J. R., *Mission Geometry; Orbit and Constellation Design and Management*, Springer, 2001.

[34] Brown, C., *Spacecraft Mission Design*, AIAA, 1998.

[35] Wertz, J. and Larson, W., *Space Mission Analysis and Design*, Microcosm Press, 1999.

[36] Netzband, M., Stefanov, W. L., and Redman, C., *Applied Remote Sensing for Urban Planning, Governance and Sustainability*, Springer, 2007.

[37] Whelan, D., Filip, A., Koss, J., Kurien, T., and Pappas, G., "Global space-based ground surveillance: mission utility and performance of Discoverer II," *Aerospace Conference Proceedings, IEEE*, , No. 5, 2000, pp. 1–11.

[38] Fortescue, P., Stark, J., and Swinerd, G., *Spacecraft Systems Engineering*, Wiley, London, UK, 2003.

[39] Guelman, M. and Kogan, A., "Electric Propulsion for Remote Sensing from Low Orbits," *Journal of Guidance, Control and Dynamics*, , No. 2, March-April 1999.

[40] Englander, J. A., Conway, B. A., and Wall, B. J., "Optimal strategies found using genetic algorithms for deflecting hazardous near-earth objects," *CEC'09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, USA, 2009, pp. 2309–2315.

[41] Abdelkhalik, O. and Mortari, D., "Reconnaissance Problem Using Genetic Algorithms," *Space Flight Mechanics Meeting Conference*, No. AAS 05–184, Copper Mountain, Colorado, January 23–27 2005.

[42] Abdelkhalik, O., "Initial Orbit Design from Ground Track Points," *Journal of Spacecraft and Rockets, AIAA*, Vol. 47, No. 1, January-February 2010, pp. 202–205.

[43] Kim, H.-D., Jung, O.-C., and Bang, H., "A computational approach to reduce the revisit time using a genetic algorithm," *International Conference on Control, Automation and Systems*, October 2007, pp. 184 – 189.

[44] Kim, H.-D., Bang, H., and Jung, O.-C., "A Heuristic Approach To The Design Of an Orbit For A Temporary Reconnaissance Mission Using A Few LEO Satellites," *AAS/AIAA Space Flight Mechanics Meeting*, January 2008.

[45] Williams, E., Crossley, W., and Lang, T., "Average and maximum revisit time trade studies for satellite constellations using a multiobjective genetic algorithm," *Journal of the Astronautical Sciences*, , No. 3, July 2001, pp. 385–400.

[46] Landis, G., "Reinventing the Solar Power Satellite, NASA/TM2004-212743, IAC02R.1.07," *Fifty Third International Astronautical Congress*, Houston, Texas, 2002.

[47] Summerer, L. and Ongaro, F., "Advanced Space Technology for 21st Century Energy Systems: Solar Power from Space," *RAST05*, Istanbul, 2005, pp. 16–23.

[48] Blandow, V., Schmidt, P., Weindorf, W., Zerta, M., Zittel, W., Bernasconi, M., Collins, P., Nordmann, T., Vontobel, T., and Guillet, J., “Earth and Space Based Power Generation Systems - A Fair Comparison for Long-Term Sustainable Energy Supply,” *European Space Agency*, GSP (17682/03/NL/EC), 2005.

[49] Edwards, B., “The Space Elevator, NIAC Phase I study,” `http://www.niac.usra.edu/studies/472Edwards.html`, 2000, Eureka Scientific.

[50] Edwards, B., “The Space Elevator, NIAC Phase II study,” `http://www.niac.usra.edu/studies/521Edwards.html`, 2000, Eureka Scientific.

[51] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press and Kluwer Academic Publications, 2nd ed., 2004.

[52] Curtis, H., *Orbital Mechanics for Engineering Students*, Elsevier Butterworth-Heinemann, Burlington, MA, 2005.

[53] Zimmerman, D. C., “GENETIC Matlab Toolbox, Software Package, Ver. 2.1,” *University of Houston, TX*, 1998.

[54] Beasley, D., Bull, D. R., and Martin, R. R., “A Sequential Niche Technique for Multimodal Function Optimization,” *Evolutionary Computation*, Vol. 1, No. 2, 1993, pp. 101–125, doi:10.1162/evco.1993.1.2.101.

[55] Izzo, D., and Vink, T., “GTOP ACT Trajectory Database, ESA,” `http://www.esa.int/gsp/ACT/inf/op/globopt.htm`, Sep. 2010, Online Database.

[56] Wertz, J. R., *Mission Geometery; Orbit and Constallation Design and Management*, Space Technology Library, 2001.