2018

System Architecture Optimization Using Hidden Genes Genetic Algorithms with Applications in Space Trajectory Optimization

Shadi Ahmadi Darani
Michigan Technological University, sahmadid@mtu.edu

SYSTEM ARCHITECTURE OPTIMIZATION USING HIDDEN GENES GENETIC ALGORITHMS WITH APPLICATIONS IN SPACE TRAJECTORY OPTIMIZATION

By

Shadi A. Darani

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Mechanical Engineering-Engineering Mechanics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2018

This dissertation has been 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

Dissertation Advisor:	*Dr. Ossama Abdelkhalik*
Committee Member:	*Dr. Nilufer Onder*
Committee Member:	*Dr. Nina Mahmoudian*
Committee Member:	*Dr. Mo Rastgaar*
Department Chair:	*Dr. William W. Predebon*

Dedication

To my parents Shahla and Mohammad, my sister Farzaneh, and my brother Farshad.

Contents

List of Figures

List of Tables

Preface

The publications in this dissertation are part of the research carried out during my PhD studies at Michigan Technological University during 2015-2018. The optimization algorithms developed in this dissertation can be used in various problems in robotics, aerospace, electrical, etc. fields. Some examples of these applications are provided in **Section** 1.1 and some mathematical and space trajectory optimization problems are investigated in details in **Chapter** 4.

Chapter 1 presents the overview of the dissertation, motivations and objective of the work, and the organization of this thesis. The content of this chapter has been published in References [4, 5]. **Chapter** 2 presents the concept of hidden Genes Genetic Algorithm in biology and optimization. The content of this chapter has been published in References [4, 6, 7]. **Chapter** 3 presents the proposed mechanisms for selecting the hidden genes. The content of this chapter has been published in References [6, 8]. **Chapter** 4 presents mathematical test cases and three space trajectory optimization problems. The mechanisms proposed in chapter 3 are tested on these problems and the results are compared to the literature. The content of this chapter has been published in References [4, 5, 6, 7]. **Chapter** 5 presents the Markov Chain convergence analysis of the proposed mechanisms and it is proven that the proposed mechanisms satisfy minimum conditions for convergence. The content of this chapter

has been published in Reference [8].

Acknowledgments

I would like express my sincere gratitude to Dr. Ossama Abdelkhalik for his continuous support, advice, and encouragement.

I with to thank the members of my dissertation committee, Dr. Nina Mahmoudian, Dr. Mo Rastgaar, and Dr. Nilufer Onder for generously offering their time, advice, and guidance throughout the preparation of this dissertation.

I would also like to thank National Science Foundation for supporting this research. The work of this dissertation was funded by National Science Foundation, award #1446622.

Superior, a high-performance computing cluster at the Michigan Technological University, was used in obtaining the results presented in this dissertation.

List of Abbreviations

GA	Genetic Algorithm
HGGA	Hidden Genes Genetic Algorithm
CGA	Canonical Genetic Algorithm
VSDS	Variable-size Design Space
DSM	Deep Space Maneuver
MGADSM	Multi Gravity-assist Deep Space Maneuver
EVM	Earth-Venus-Mars
EVEJ	Earth-Venus-Earth-Jupiter
DNA	Deoxyribonucleic Acid
TOF	Time of Flight
ESA	European Space Agency
PaGMO	Parallel Asynchronous Generalized Island Model Optimization

Abstract

In this dissertation, the concept of hidden genes genetic algorithms is developed. In system architecture optimization problems, the topology of the solution is unknown and hence, the number of design variables is variable. Hidden genes genetic algorithms are genetic algorithm based methods that are developed to handle such problems by hiding some genes in the chromosomes. The genes in the hidden genes genetic algorithms evolve through selection, mutation, and crossover operations. To determine if a gene is hidden or not, binary tags are assigned to them. The value of the tags determine the status of the genes. Different mechanisms are proposed for the evolution of the tags. Some mechanisms utilize stochastic operations while others are based on deterministic operations. All the proposed mechanisms are tested on mathematical and space trajectory optimization problems. Moreover, Markov chain models of the mechanisms are derived and their convergence is investigated analytically. The results show that the proposed concept are capable to search for the optimal solution by autonomously enabling the algorithms to assign the hidden genes.

Chapter 1

Introduction

1.1 Overview

Systems architecture optimization problems [1] arise in several applications such as in automated construction (in which hundreds or thousands of robots fabricate large, complex structures), autonomous emergency response, and smart buildings, transportation, medical technology, and electric grids [9]. In these complex systems, the automated system design optimization is crucial to achieve design objectives. The task of design optimization includes optimizing the system architecture (topology) in addition to the system variables. Optimizing the system architecture renders the problem a Variable-Size Design Space (VSDS) optimization problem (the number

[1] The material of this chapter are copied in part from References [4, 5]

of design variables to be optimized is a variable). Consider, for example, the optimization of a space interplanetary trajectory. The spacecraft travels from the home planet to the target planet and it is desired to utilize the minimum fuel possible. As can be seen in **Figure** 1.1, the spacecraft can apply Deep Space Maneuvers (DSMs) which are propulsive impulses used to change the velocity of the spacecraft instantaneously; these DSMs consume fuel proportional to the amount of the DSMs impulse. The spacecraft can also benefit from free change in momentum, through as many as needed flybys of other planets. When the spacecraft performs a flyby maneuver, we need to determine the height of closest approach to the flyby planet as well as the plane of the flyby maneuver. Hence, by changing the number of flybys the total number of variables change.

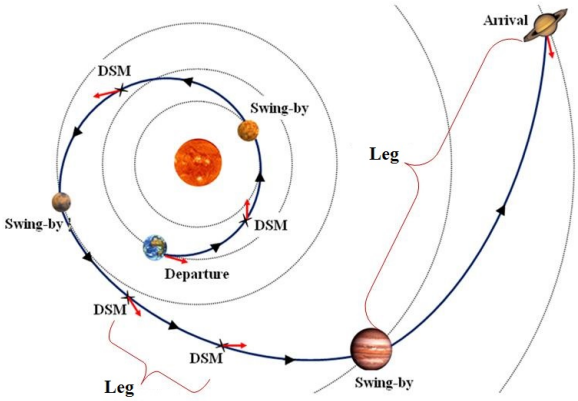


Figure 1.1: Interplanetary Trajectory Optimization Problem Topology

Besides the flyby planets, the spacecraft can have DSMs in any segment between any two planets. These segments are referred to as legs. The architecture of a solution refers to the sequence of flybys and the number of DSMs in each leg. To optimize

the mission architecture, the number of flybys, the planets of flybys, and the number of DSMs in each leg need to be optimized. These are called the architecture Other non architecture variables include launch and arrival dates, dates and times of flybys, dates and times of DSMs, amounts and directions of DSMs impulses. This is a VSDS optimization problem.

Another example is the optimization of a microgrid system where there are several energy sources and co-located energy storage devices that can either sink or source power with their corresponding sources. The net power at each source/storage is metered to the grid main bus using a boost converter. For an efficient design of the microgrid, the number of storage elements (N) and their capacities need to be optimized. Storage is expensive and designing a microgrid, with storage sized properly, is an open problem. Associated with computing the optimal N is the optimal values for the duty ratios at the converters that controls the power metered to the main bus from each source. A more complex situation is when we have M microgrids that have the ability to interconnect. This provides a large number of permutations for exchanging power. Systems design optimization problems are usually replete with local minima. Hence a global search algorithm is usually needed for optimizing the system variables, such as genetic algorithms (GAs) [10], particle swarm optimization [11], ant colony optimization [12], or differential evolution [13]. In VSDS optimization, the problem

can be formulated as follows:

$$\begin{aligned} &\text{Minimize} \quad f(\vec{x}, N) \\ \text{Subject to} \quad &\vec{g}(\vec{x}) \leq \mathbf{0}, \quad \vec{h}(\vec{x}) = \mathbf{0}, \quad \vec{x}^l \leq \vec{x} \leq \vec{x}^u \end{aligned} \tag{1.1}$$

where $\vec{x} = [x_1, x_2, ..., x_N]^T$, N is the number of design variables, $\vec{x}^u$ and $\vec{x}^l$ are the upper and lower bounds of the variables $\vec{x}$, respectively. The number of variables N in this formulation is variable, and its value dictates the architecture of the solution. The number of inequality constraints $\vec{g}$ and the number of equality constraints $\vec{h}$, each is also a variable.

1.2 Genetic Algorithms

The research on developing algorithms that can handle VSDS optimization problems (sometimes referred to as variable length optimization) has started since about two decades. Standard GAs are not suitable for VSDS problems because they are designed to work only on problems of fixed number of variables. In standard GAs, the variables of the optimization problem are coded in chromosomes. Each chromosome represents a solution and consists of the variables that are coded as genes. In standard GA, the number of variables is assumed fixed and therefore, the number of the genes and the length of the chromosomes are fixed. By applying the evolutionary operations of

selection, mutation, and crossover, the population of these chromosomes converges to the global optimal solution [10]. The objective of optimization determines the fitness of the solution. The genetic operations of selection, mutation and crossover are applied on a population of these chromosomes, and through generations (iterations), theses populations converge to the optimal solution.

In the selection operation, two chromosomes are selected as parents from the generation pool. In general, the chromosomes that have better fitness values (objective function), have higher probability to be selected as parents. After the parents are selected, mutation and crossover operations are applied on them. As an example, in binary coding, genes are coded as 0s and 1s; in the mutation process gene 1 may change to 0 with a probability of p_m. In the crossover operation, parts of the chromosome strings are swapped in parents. For example, in single point crossover, a random point is selected in both parents and the genes of one side of that point are swapped in parents with a crossover probability of p_c to create new chromosomes. Some of the best chromosomes (elites) are transferred to the next generation with no change. By repeating the GA operations in each generation, the population converges to the optimal solution.

Some variations of GAs have been proposed for VSDS problems. Genetic programming is a specialization of genetic algorithm in which each individual is a computer program [14, 15]. In genetic programming, the solutions are in the structure of trees

that can have variable lengths. A VSDS GA is presented in [16] in which a random operator is introduced to change the chromosome length, for the problem of Kauffman NK model. This random operator depends on the identity of genes which is given by their position relative to one end of the genotype. Reference [17] is a continuing work of [16] and analyzes the optimal location for the crossover point in VSDS problems. When two parents have different chromosome lengths, and given a selection for the crossover point in parent 1, reference [17] suggests that the crossover point in parent 2 be chosen such that the difference between the swapped segments is minimized. The method proposed in [17] is a search on all the possible crossover points in parent 2 to find the best cutoff point. The VSDS GA in reference [18] uses a two-point crossover, with different cutoff points in each parent, resulting in different lengths of the children chromosomes. This method is most useful in problems with variables of the same identity, like angles of a polyhedral where adding or removing one angle will result in a new polyhedral (e.g. triangle to rectangle or vice versa).

Reference [19] presents a number of variable length representation evolutionary algorithms that improves the sampling of a VSDS, with application in evolutionary electronics. In reference [20], the number of different chromosome lengths is set a priori, and both parents have the same crossover point (same gene index of cutoff). Therefore the length of the chromosome is switched from parents to children in [20] (the length of child 2 is equal to length of parent 1 and length of child 1 is equal to length of parent 2). This method does not provide information regarding the optimal

length of a solution. A different approach in VSDS GA is to have equal-length chromosomes in each generation, yet the chromosome length is allowed to change among different generations as presented in [21, 22]. In this method, the GA starts with short-length chromosomes and the best solution in a generation is transferred to the next generation with a longer chromosome length. In this way, the GA handles fixed-size chromosomes in each generation, and there is no need to define new evolutionary operations for GA.

A structured chromosome genetic algorithm was developed in [23, 24] where the standard one layer chromosome is replaced with a multi-layer chromosome for coding the variables; the number of genes in one layer is dictated by the values of some of the genes in the upper layers. Hence, it was possible to code solutions of different architectures. Yet, this structured-chromosome approach introduces new definitions for the crossover operation such that meaningful swapping between chromosomes of different layers is guaranteed. Some other algorithms are designed for specific problems. For instance, references [25] and [26] present tailored algorithms that search for the optimal structural topology in truss and frame structures, respectively. The dissertation in [27] presents a study on topology optimization of nanophotonic devices and makes a comparison between the homogenization method [28] and genetic algorithms [10]. As can be seen from the above discussion, many of the VSDS optimization algorithms are problem specific. The dynamic-size multiple population genetic algorithm has a high computational cost [29].

1.3 Space Trajectory Optimization

Space trajectory optimization is the process of searching for the optimal trajectory from one celestial body or orbit to another, such that the mission requirements are satisfied and a given objective is optimized. The objective can be minimizing the mission cost or fuel consumption, minimizing the mission duration, maximizing the number of visited asteroids, or a combination of these objectives. The spacecraft can have continuous or impulsive thrusters, for which various trajectory design techniques have been developed. In this dissertation, the impulsive thrust spacecraft is considered. The earliest research on space trajectory optimization goes back to the work of Walter Hohmann on trajectory design of a spacecraft with impulsive thrusters between two coplanar orbits [30]. Cornelisse [31] showed that in the patched conics method, the cost of an interplanetary trajectory mission can be reduced by applying a DSM. Several works have studied the effect of DSMs in different space missions [32, 33, 34, 35]. Planetary flybys utilize the gravity of a planet to change the momentum vector of a spacecraft. Such trajectories that use DSMs and flybys are called Multi Gravity-Assist Deep Space Maneuver (MGADSM) trajectories.

To design a MGADSM interplanetary trajectory, many variables should be optimized depending on the mission type, such as launch and arrival dates and times, number of

flybys, planets to flyby, number of DSMs, epoch of each DSM, direction and magnitude of each DSM, time of flight (TOF) between each two successive celestial bodies (leg), and flyby altitudes and rotation angles. These variables can be categorized into two groups of discrete design variables and continuous design variables, as shown in **Table** 1.1. Since some variables are related to others (e.g. flyby attitude depends on whether there is a flyby or not), the problem can be considered as VSDS problem, in which the number of optimization variables vary among different trajectory missions. In other words, the number of flybys and DSMs are not known a priori and they determine the number of other variables needed to model the problem. These variables that determine the total number of variables in a solution are referred to as the architecture variables.

Table 1.1
Design variables in an interplanetary trajectory optimization problem

Discrete Variables	Continuous Variables
Number of flybys (m)	Departure date (t_d)
Flyby planets (P)	Arrival date (t_a)
Number of DSMs in each leg (n)	TOF
	Flyby pericenter altitude (h_p)
	Flyby rotation angles (η)
	DSMs epoch (ϵ)
	DSMs magnitudes and directions

Many global optimization methods have been investigated in different MGADSM

problems, including heuristic algorithms [36, 37, 38, 39, 40, 41], deterministic algorithms [42, 43, 44], or a combination of them [45, 46]. Deterministic methods use grid or tree search to explore the design space. Although these methods converge globally, they can be extremely exhaustive, especially in complex missions with high number of flybys/rendezvous and DSMs or large time windows. The obtained solutions also are usually sensitive to the grid size. Heuristic methods on the other hand do not need discretization of the search space and are more adaptive and hence are not usually exhaustive. Yet they rely on heuristics and parameters tuning. Genetic algorithms (GAs) [47, 48, 49, 50, 51], differential evolution [41, 52, 53, 54], and ant colony optimization [55] are some of the heuristic algorithms that have been used in MGADSM optimization problems.

In MGADSM problems, since in general the flyby and DSM structures are not known a priori, it is not possible to use the standard GAs without simplifications on the problem or modification to the algorithms [36, 56, 57]. One way of simplifying the problem is to prune the state space (assume fixed flyby sequence and number of DSMs) to limit the possible mission scenarios. A deterministic search is used in [57, 58] where the flyby sequence and DSMs are fixed and the search space is limited to a grid of points where the global optimization methods can be used. Another way of simplifying the problem is to use a nested loop solver to optimize the trajectory [37, 59]. The outer loop finds the optimal flyby sequence and the inner loop optimizes the trajectory for that scenario. Since not all the scenarios have the same number

of flybys, this problem is a VSDS optimization. Early methods pruned the outer loop to solutions that the designer considered to include the optimal flyby sequence [60]. Later, automatic methods were proposed to find the flyby sequence in MGA trajectories. Some graphical methods use the energy contours against two variables that define the orbits for different planet flybys [61, 62]. This method can be used when all the flybys are considered non-powered and it is assumed that there is no DSMs. In [59] a maximum length for the flyby sequence is assumed and the outer loop is optimized using a binary genetic algorithm. By adding null variables that represents a "no flyby", variable-sized flyby sequences can be modeled in this method. For example, for a maximum flyby of two, the Earth-Venus-Mars (EVM) sequence is equivalent to a mission from Earth to Mars with a flyby around Venus and a null flyby that is not considered in the cost function. Genetic algorithm is also used for multiple phase maneuvers where there is both impulsive and continuous maneuvers [37]. Genetic Programming [15] is also among the earliest approaches that addressed the VSDS optimization problems. One of the earliest attempts in implementing gene expression in GA is to perform "cut and splice" on the chromosomes and applying a self adaptive recombination operator on them to yield individuals of variable lengths [63, 64]. In recent years, the role of histone in the regulation of DNA including gene expression and functionality of each cell was discovered [65], which resulted in the use of epigenetics through modification of histone in strongly-typed genetic programming [66]. A dynamic-size multiple population genetic algorithm was developed in [29]

where each generation consists of a number of sub-populations; all chromosomes in each sub-population are of the same length. Hence each sub-population evolves over subsequent generations as in a standard GA. The size of each sub-population, however, changes dynamically over subsequent generations such that more fit sub-populations are allowed to increase in size whereas lower fit sub-populations decreases in size. This approach has been applied to the trajectory optimization problem and demonstrated success in finding best know solution architectures. The computational cost of this method, however, is relatively high since it implements GA over several sub-populations in parallel. Also, only a finite number of architectures (assumed a priori) can be investigated using the method in [29].

1.4 Motivations and Objectives

Inspired by the concept of gene expression in biology, the concept of Hidden Genes Genetic Algorithm (HGGA) was introduced to search for the optimal architecture and autonomously generate new design spaces [2, 3]. Reference [3] applied a simplified version of the HGGA for interplanetary trajectory optimization and demonstrated success in finding the best known solution architectures for known benchmark problems. This original version of the HGGA implemented in [3] assumes a long chromosome for each solution where some of the genes are hidden. In this version, genes in a chromosome will only be hidden if a chromosome represents a non-feasible solution,

Hence, the HGGA will not attempt to hide genes if the chromosome is a feasible solution. Therefore, this developed method of HGGA lacks a rigorous mechanism for selecting the hidden genes in each generation.

In this dissertation, the objective is to develop new mechanisms for hiding genes. These mechanisms should have the following properties:

1. They should autonomously decide which genes should be hidden.

2. They should not be problem specific; i.e. they should be applicable to any VSDS problem with no modification of the algorithm in any kind or simplification on the problem. The problems can be constrained or non-constrained, discontinuous, non-differentiable, stochastic, or highly nonlinear.

3. They must promise convergent solutions.

4. They should produce comparable results to other algorithms for VSDS problems.

Note that the proposed algorithms in this dissertation are stochastic optimization algorithms. Hence, there is no guarantee that the solutions found with these algorithms are global optima. In this dissertation however, the results found by the designed algorithms are referred to as the *optimal points/solutions*.

1.5 Organization of the Dissertation

This dissertation consists of six chapters covering the objectives described in the previous section. In preparing this dissertation, the materials of papers written during this research are utilized. **Chapter** 2 covers the necessary background material on GAs and HGGAs. The concept of HGGAs in biology is explained and the original feasibility criteria for HGGAs is described. **Chapter** 3 is focused on developing new HGGA mechanisms. The concept of tags and alleles in HGGAs are presented and several stochastic and deterministic evolution mechanisms of the tags and alleles are proposed. In **Chapter** 4, the performance of these mechanisms are tested on various VSDS problems, including mathematical problems and space trajectory optimization problems. In **Chapter** 5, the Markov Chain convergence analyses of the HGGAs with the proposed mechanisms are performed and finally the conclusion of the dissertation is presented in **Chapter** 6.

Chapter 2

Hidden Genes Genetic Algorithms

2.1 Introduction

In this chapter [1], the concept HGGAs is presented. To handle a VSDS (or architecture) optimization problem, the idea of turning genes on and off was adapted from biology in genetic algorithm. By setting the chromosome length equal to the length of the longest possible chromosome L_{max} (maximum number of design variables) and turning some genes off, different solutions (of different architectures) with lengths of 1 to L_{max} can be built while having the same length for all the chromosomes. Moreover, having similar lengths for all the chromosomes enables the implementation of the standard GA operations like crossover and mutation on them. The genes that are

[1] The material of this chapter are copied in part from References [4, 6, 7]

hidden are variables that do not affect the fitness of the solution; yet they carry information, go through GA operations, and may become active (not hidden) in future generations. In the next section, the equivalent of HGGAs in biology is explained and an initial hidden genes assignment mechanism is described.

2.2 Modeling Genes and Chromosomes

In standard GAs, the mechanics of natural selection and genetics are simulated [10]. For each solution a chromosome is considered which is a set of coded variables called genes. **Figure** 2.1 shows a typical chromosome that consists of N genes $g_1, g_2, \ldots, g_N$. The value of g_i determines the value of that variable in that solution. The fitness of the solution is determined based on the objective of optimization.

Figure 2.1: In standard GA, a chromosome (code) is a string of genes that represent a solution

The algorithm starts by applying the genetic operations of selection, crossover, and mutation on a population of these chromosomes. Through generations (iterations), this population converges to optimal solutions. In the selection operation, the chromosomes that are more fit, have higher probability of being selected as parents. After parents are selected, the crossover and mutation operations are applied on them to

create new children chromosomes. For example, in the single point crossover, a random point in parents strings is selected and the gene strings of both sides of that point are swapped in parents to create new individuals. The crossover probability of p_c is applied to the crossover operation to make sure that the fit individuals found in the previous population survive without modification. In the mutation operator, each gene is mutated with probability of p_m. For example, in binary coding, gene 0 may change to 1 through mutation operator. By repeating the selection, crossover, and mutation operations in each generation, the population converges to a (near) optimal solution.

2.3 Hidden Genes in Biology

In genetics, the deoxyribonucleic acid (DNA) is organized into long structures called chromosomes. Contained in the DNA are segments called genes. Each gene is an instruction for making a protein. These genes are written in a specific language. This language has only three-letter words, and the alphabet is only four letters. Hence, the total number of words is 64. The difference between any two persons is essentially because of the difference in the instructions written with these 64 words. Genes make proteins according to these words. Since, not all proteins are made in every cell, not every gene is read in every cell. For example, an eye cell doesn't need any breathing genes on. And so they are shut off in the eye. Seeing genes are also shut

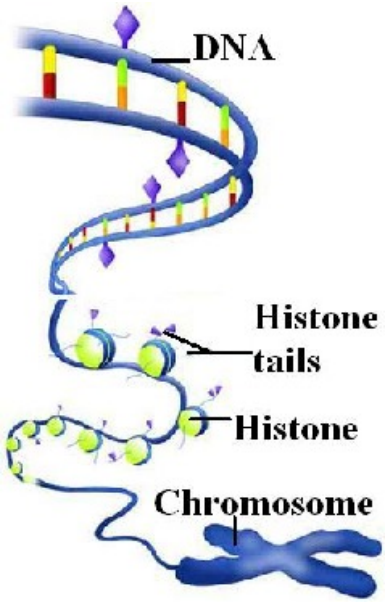


Figure 2.2: Chemical tags (purple diamonds) and the "tails" of histone proteins (purple triangles) mark DNA to determine which genes will be transcribed. (picture is modified from [1])

off in the lungs. Another layer of coding tells what genes a cell should read and what genes should be hidden from the cell [67]. A gene that is being hidden, will not be transcribed in the cell. There are several ways to hide genes from the cell. One way is to cover up the start of a gene by chemical groups that get stuck to the DNA. In another way, a cell makes a protein that marks the genes to be read; Figure 2.2 is an illustration for this concept. Some of the DNA in a cell is usually wrapped around nucleosomes but lots of DNA are not. The locations of the nucleosomes can control which genes get used in a cell and which are hidden [67].

2.4 Concept of HGGAs in Optimization

The concept of Hidden genes is applied in GA to make some genes hidden (inactive), so that their value does not affect the fitness of objective function. The genes that are hidden are variables that should not appear in a specific solution. This concept allows GA to be able to handle VSDS and architecture optimization problems. In such problems, the number of design variables is variable and the length of the chromosome changes by selecting different values for some of the design variables. Let L_{max} be the length of the longest possible chromosome (maximum number of design variables). In hidden gene concept, all the solutions (chromosomes) have the same length and hence the operators of standard GA can be applied to them. Genes that are hidden will be ineffective in fitness of the objective function, although they take part in the genetic operations in generating future generations.

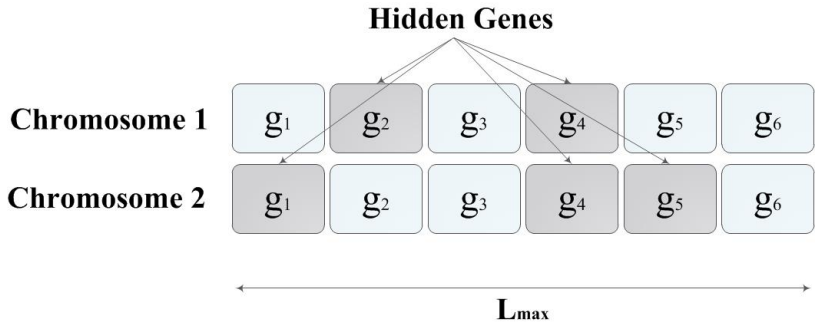


Figure 2.3: Hidden genes and effective genes in two different chromosomes [2]

Consider two chromosome with different lengths. Assume that there are four genes in the first chromosome and three genes in the second chromosome (represented by binary bits in **Figure** 2.3). Also assume that the maximum number of genes (variables) in a chromosome is six. To make this problem a fixed-sized design space problem, two hidden genes are added to the first chromosome, and three hidden genes are added to the second chromosome. These added genes are hidden and therefore do not affect the fitness of the objective function. Since all the chromosome have the same length now, the standard GA operators can be applied to them. These added (hidden) genes go through crossover and mutation like active genes. Based on the mechanism that assign the hidden genes, a hidden gene in parents can be active in children (and hence effective in fitness evaluation).

A simple example of a single-point crossover operator in HGGA is shown in **Figure** 2.4. In this figure, the crossover point is between the second and third genes. After the genes are swapped, the location of hidden genes in children may be similar to or different than the hidden genes in the parents. The genes that should be hidden are selected based on a specific hidden gene assignment method. In the initial studies on HGGA [2, 3], a primitive mechanism (called *feasibility* mechanism") was introduced. In these works, the hiding criteria was feasibility, meaning that the genes were all active unless the solution was infeasible. In that case, the genes would be hidden one by one until a feasible solution is achieved.

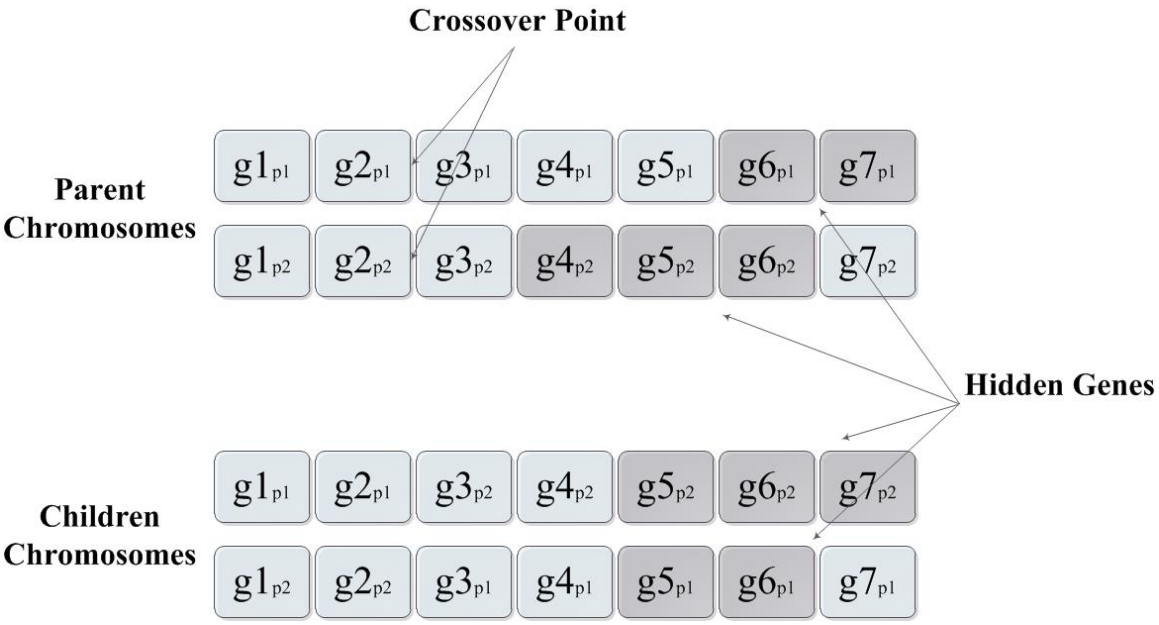


Figure 2.4: Crossover operation in HGGA [2]

2.5 Conclusion

The concept of hidden genes and their equivalent in biology were introduced in this chapter. To determine the status of genes, the feasibility mechanism can be utilized where genes are hidden one by one from one side of the chromosome until a feasible solution is acquired. However, this mechanism is not robust specially when variables are simulated in various locations of the chromosome as genes. In the next chapter, new mechanisms are proposed for hiding gens. In these mechanisms, the status of the genes can evolve through generations while the genes are evolving.

Chapter 3

Hidden Genes Assignment Methods

3.1 Introduction

In this chapter [1], new mechanisms are proposed to hide genes. The concept of tags is described and eight stochastic mechanisms with tags, three logical mechanisms, and the concept of alleles are presented. In the logical mechanisms, the logical *OR* and *OR* operators are used. For x and y expressions, the logical OR operator is true if either of x or y are true. The logical AND operator results in true if x and y are true.

[1]The material of this chapter are copied in part from References [6, 8]

As described in **Section** 2, the protein of each gene makes it to be read or hidden. This gave us the idea to use tags for genes to make them hidden or active. To code such tags, binary digits of 0 and 1 are assigned to each tag, as shown in **Figure** 3.1. If the value of tag_i is 1, then the corresponding gene x_i is hidden, and if it is 0, gene x_i is not hidden (active).

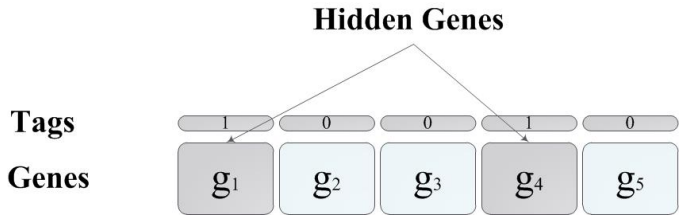


Figure 3.1: HGGA and the tags concept

Chromosomes evolve over successive generations. Genes along with their tags go through evolutionary operations. Genes evolve through the standard operations defined in the CGA. The tags, however, may evolve with different operations. A set of operations used to evolve tags is here referred to as a mechanism for tags' evolution.

3.2 HGGAs Mechanisms

There are 12 different mechanisms for tags evolution that will be investigated in this section. In the mechanisms that have a crossover operator for the tags, the singe-point crossover is used, unless otherwise stated. Some of the evolution mechanisms are logical. Here we introduce two definitions. Consider two parents selected for

reproduction and consider one offspring child. The Hidden-OR evolution logic is defined as follows: a gene in the child chromosome is hidden if the same gene is hidden in any of the parents. The Active-OR evolution logic is defined as: a gene is active in the child if the same gene is active in any of the parents.

1. Mechanism A: tags evolve using a crossover operator. The crossover point location in the tags can be different from that in the genes. Before the crossover, tags go through a mutation with probability of 10%.

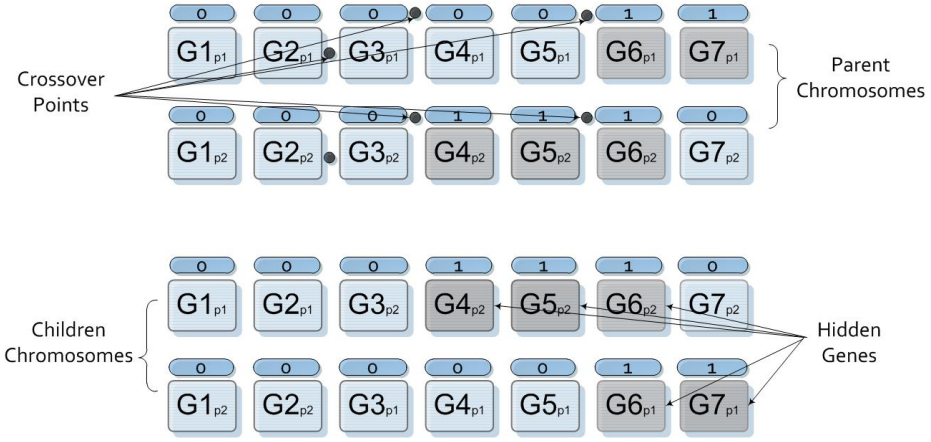


Figure 3.2: Schematic of Mechanism A

2. Mechanism B: When two parents are selected for reproduction, then the process of evolving the tags is as follows:

   i - produce two temporary children through a single-point crossover operation on genes, and an Active-OR logic on tags. Both of these temporary children will have the same tags.

   ii - calculate the fitness value of these two temporary children, $\bar{f}_1$ and $\bar{f}_2$.

iii - consider the parents chromosomes (genes and tags) as points in $\mathbb{R}^{L+L_t}$ space where L_t is the number of tags.

iv - the child (output of Mechanism B) is the weighted arithmetic crossover on the parents and is closer to the parent that has better fitness $\bar{f}$ for its temporary child.

For example, for the $\mathbb{R}^3$ space in **Figure** 3.3, the child is closer to parent 1 because its temporary child has better fitness value. λ is a random number in $(0, 0.5)$. If $\bar{f}_1 = \bar{f}_2$, then the child can be randomly closer to either parents.

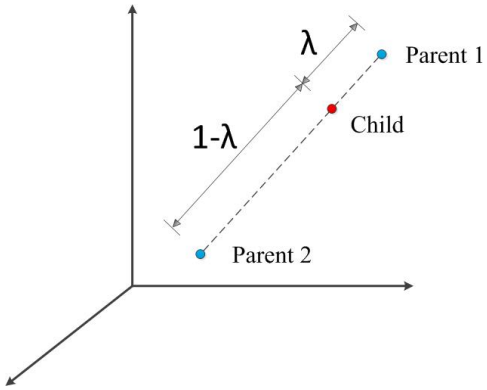


Figure 3.3: Representation of arithmetic crossover in $\mathbb{R}^3$.

In this mechanism, the mutation operator is only allied to the genes.

3. Mechanism C: The arithmetic crossover operator is used for the genes only. The tags in the child will have the same tags of one of the parents depending on the value of $\left(f_{m_1} = f + \sum_{i=1}^{L_t} tag_i\right)$, where f is the fitness of the parent. The offspring tags will be the same as that of the parent that has better value of $\left(f_{m_1} = f + \sum_{i=1}^{L_t} tag_i\right)$. In other words, this mechanism favors higher number

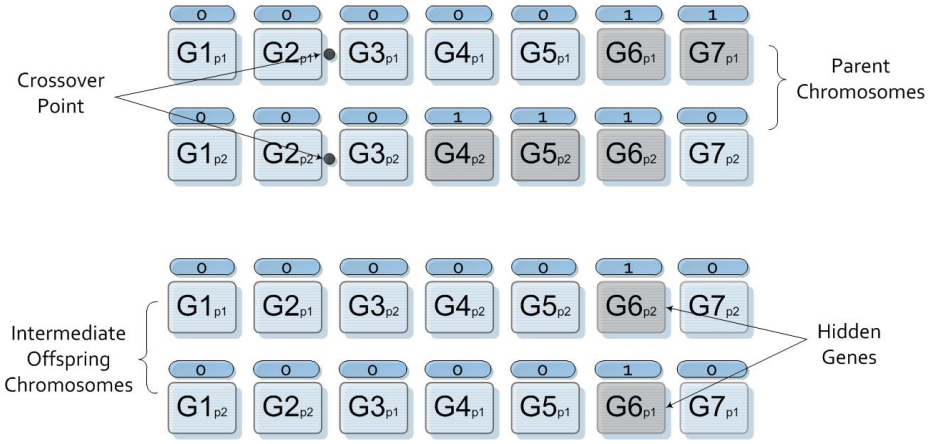


Figure 3.4: Schematic of Mechanism B.

of hidden genes.

4. Mechanism D: same as Mechanism C, but the offspring tags have the same values as that of the parent with better value of $\left(f_{m_2} = f - \sum_{i=1}^{L_t} tag_i\right)$. In other words, this mechanism favors less number of hidden genes.

5. Mechanism E: tags evolve only through a mutation operation with a certain mutation probability different than the mutation probability of the genes. So, two parents are selected; then mutation for the genes is carried out and another mutation for the tags is carried out. These two parents then go through a crossover operation on the genes with a certain probability as in the CGA, while the tags remain unchanged during this crossover operation.

6. Mechanism F: tags are considered as discrete variables where they are appended to the genes to create a long chromosome that has both genes and tags. Then the mutation and crossover operations are carried out in a similar way to that of the CGA.

7. Mechanism G: this mechanism is similar to Mechanism F except that the tags do not go through a mutation operation.

8. Mechanism H: this mechanism is similar to Mechanism F except that the tags do not go through a crossover operation. This is carried out by limiting the crossover point to be within the genes only.

9. Alleles: in biology, an allele is an alternative form of a gene and in human cells, there are two alleles of a gene in each position on a chromosome, one dominant and one recessive. Dominant traits are expressed only when the alleles of a pair are heterozygous (the individual only has one copy of the allele). For example, the allele for brown eyes is dominant, meaning that there is only one allele of brown eyes needed to have brown eyes. On the other hand, the recessive traits are expressed only if the alleles of a pair are homozygous (the individual has two copies of the allele). These principles and their traits was first discovered by Gregor Mendel [68, 69] and is named as Mendel's Law of Segregation. Knowing this concept in biology, two sets of tags (alleles) are considered for genes in HGGA, in which only the dominant allele decides whether a gene is hidden or active but both dominant and recessive alleles go through GA operations and affect the next generation's status. Therefore a recessive allele in the current generation may become a dominant allele in the next generation. For the evolution process, the mutation operation is first carried out in the genes and tags. Then, a single-point crossover operator is applied to the genes, and a two-point

crossover operator is applied to the tags such that the crossover point in the dominant and recessive tags are similar.

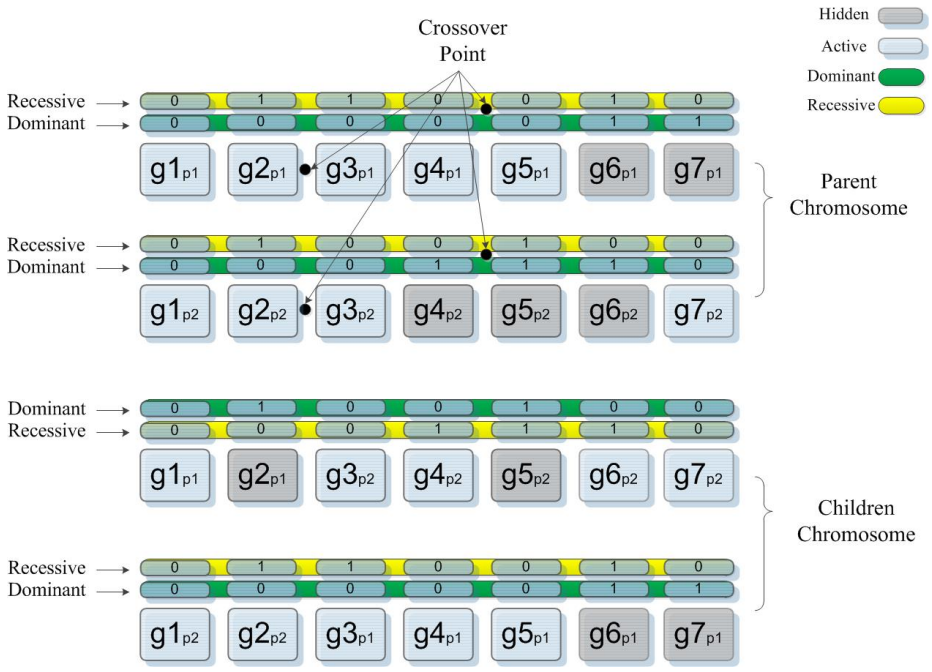


Figure 3.5: Schematic of Alleles Mechanism.

10. Logic A: the member of the current generation ($\bar{n}$) is split into two groups of equal size. For the first group, the Active-OR logic is used for tags evolution (a gene is active in the child if the same gene is active in any of the parents). For the second group, the Hidden-OR logic is used for tags evolution (a gene is hidden in the child if the same gene is hidden in any of the parents).

11. Logic B: similar to Logic A; but the Hidden-OR logic is used for all the members in the generation.

12. Logic C: similar to Logic A; but the Active-OR logic is used for all the members

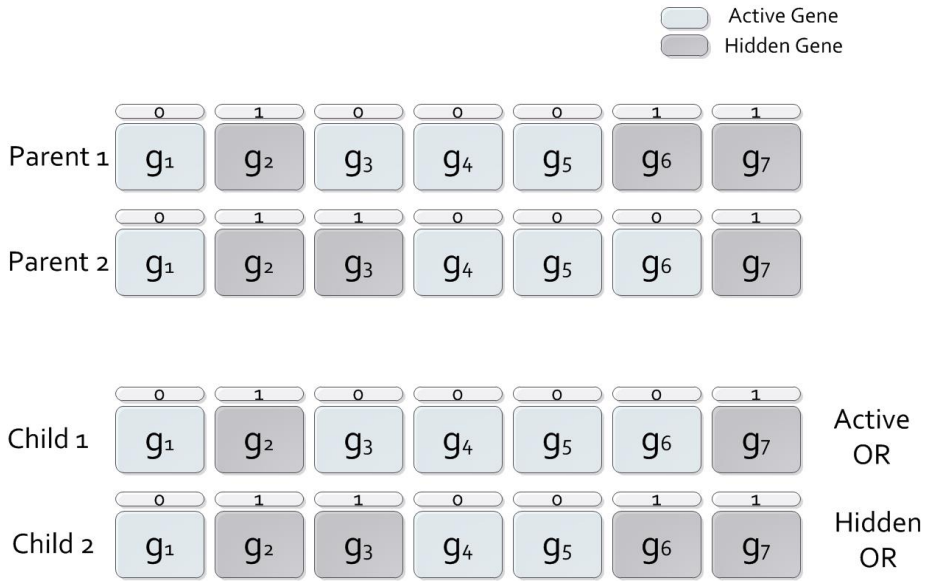


Figure 3.6: Schematic of Logic A.

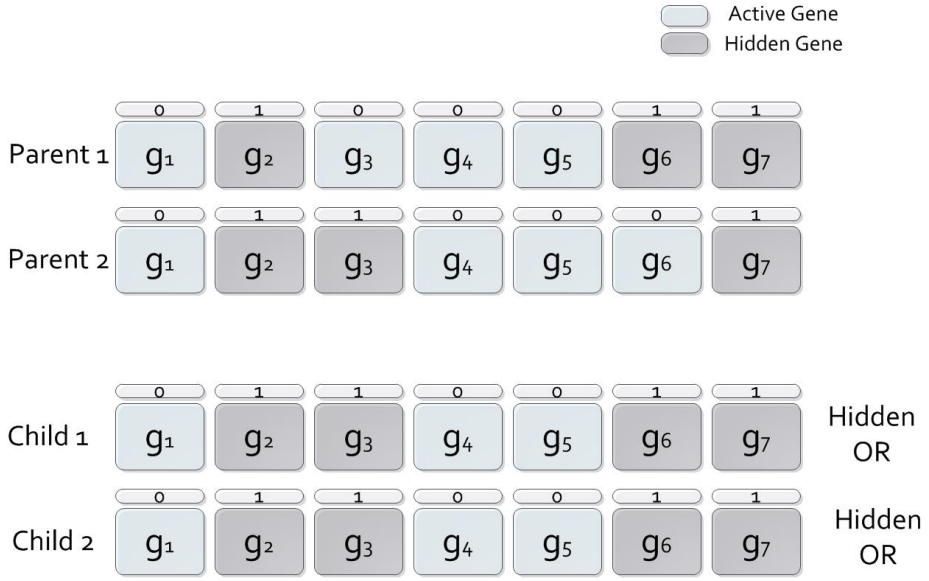


Figure 3.7: Schematic of Logic B.

in the generation.

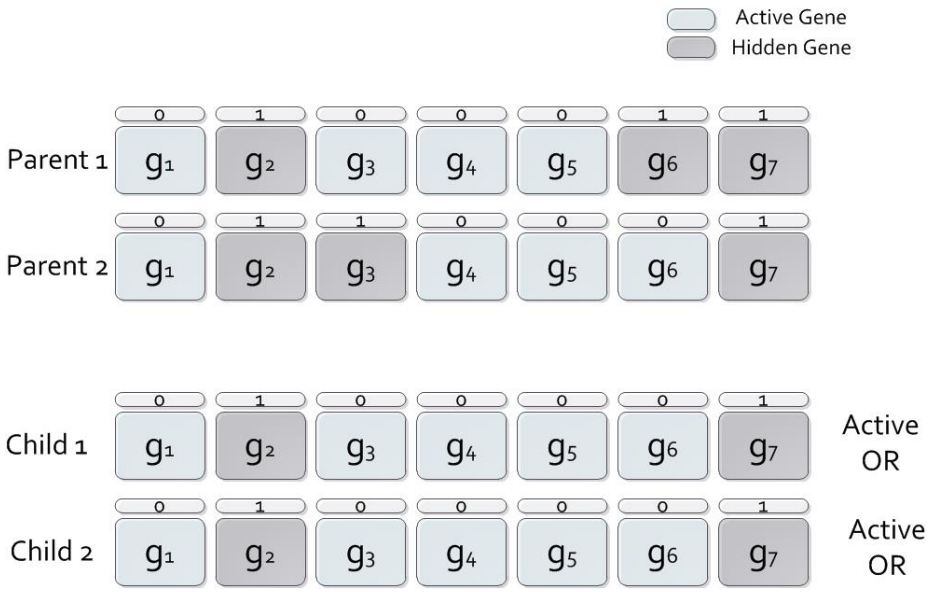


Figure 3.8: Schematic of Logic C.

3.3 Conclusion

In this chapter, the concept of binary tags was introduced in genetic algorithms to enable hiding some of the genes in a chromosome, so that they can be used to search for optimal architectures in VSDS problems. The proposed binary tags concept mimics biological cells in hiding the genes that are not supposed to be effective in the cell, while they could be effective in other cells. Various mechanisms for assigning the chromosome hidden genes were proposed and investigated in this chapter. These mechanisms make assigning the status of the genes more robust compared to the feasibility criteria and can be applied to any problem with various number of hidden genes for different types of variables. By evolving tags through generations, the status of the genes evolve at the same time that their values evolve. This extended

functionality in HGGAs is robust and easy to implement. All the mechanisms are tested on different VSDS problems in the following chapter.

Chapter 4

Test Cases

4.1 Introduction

In this chapter [1], the proposed mechanisms are tested on two types of VSDS problems. The first type is mathematical problems that are used as the primary performance evaluation for the mechanisms. Their performance is compared to the initial concept of HGGAs (feasibility criteria). After that, the mechanisms are tested on three space trajectory optimization problems. These problems have different levels of complexity and are adapted from the standard space trajectory optimization benchmark. The performance of the mechanisms are compared and their capabilities are evaluated.

[1] The material of this chapter are copied in part from References [4, 5, 6, 7]

4.2 VSDS Mathematical Functions

Multi-minima mathematical functions can be very useful in testing new optimization algorithms. However, there is not many multi-minima VSDS mathematical functions in the known benchmark mathematical optimization problems. Four benchmark mathematical optimization problems were modified to make them VSDS functions; and then they were used to test the new HGGA mechanisms. These functions are: the Egg Holder, the Schwefel 2.26, and the Styblinski-Tang functions. These functions and their variable ranges are as follows [70]:

1. Egg Holder: continuous, differentiable, and multimodal.

$$F_{EG}(X) = \sum_{i=1}^{N}(-(x_{i+1}+47)sin(\sqrt{|x_{i+1}+x_i/2+47|})- \\ x_i sin(\sqrt{|x_i-(x_{i+1}+47)|})), \quad -512 \leq x_i \leq 512 \tag{4.1}$$

2. Schwefel 2.26: continuous, differentiable, and multimodal.

$$F_{Sch}(X) = -\frac{1}{N}\sum_{i=1}^{N} x_i sin(\sqrt{|x_i|}, \quad -500 \leq x_i \leq 500 \tag{4.2}$$

3. Styblinski-Tang: continuous, differentiable, and multimodal.

$$F_{Sch}(X) = \frac{1}{2}\sum_{i=1}^{N}(x_i^4 - 16x_i^2 + 5x_i), \quad -5 \leq x_i \leq 5 \tag{4.3}$$

The general concept of modifying these functions to be VSDS functions is here described. Consider the optimization cost function defined as:

$$F(X) = \sum_{i=1}^{N} f_i \tag{4.4}$$

If tag_i is 1 (hidden), then f_i is set to zero. In other words, if a variable (gene) i is hidden, then the corresponding f_i is zero, or does not exist. This is consistent with the physical systems test cases presented in **Section** 1.1. Unlike the tags, the genes evolve through the standard GA selection, mutation and crossover operations. In general, standard GA are not suitable for solving VSDS problems. However, a significant advantage of using the above modified mathematical functions is the possibility of using standard GA if we assume all variable are active (not hidden). If the optimal solution has x_j hidden $\forall j \in \Gamma$, and $\Gamma \subseteq \{1, 2, \cdots, N\}$, then the standard GA can find that optimal solution, if we assume all variables are not hidden. In such case, the optimal solution that the standard GA would search for is x_j^* where $f(x_j^*) = 0$, $\forall j \in \Gamma$.

All the proposed mechanisms are tested on the selected mathematical optimization

functions. For the genes, a single point crossover and an adaptive feasible mutation operators are selected. The GA parameters used in these simulations are listed in **Table** 4.1.

Table 4.1
Genetic Algorithm Options

Option	*Value*
Population Size	400
Number of Generation	300
Elite Count	20
StallGenLimit Count	25
Crossover Fraction	0.95
TolFun	$1e-6$

where *Elite Count* is the number of solutions that go to the next generation without any change and the algorithms stops if the average relative change of the best solution over *StallGenLimit* generations is less than or equal to *TolFun*. Note that the crossover fraction in **Table** 4.1 is for the genes and the tags evolve based on the characteristics of each mechanism. In **Equation** 4.4, if f_i is a function of x_i only, there are N tags, and if f_i is a function of x_i and x_{i+1}, then there are $N-1$ tags. In all the problems, the number of variables without tags is 5. Each test case is simulated 20 times. Superior, a high-performance computing cluster at the Michigan Technological University, was used in obtaining the results presented in this section. This computing cluster is Generation 2 with 47 CPU compute nodes, each having 32 CPU cores (Intel Xeon $E5-26832.10$ GHz) and 256 GB RAM [2]. The best solutions,

[2]MTU High Performance Computing, https://hpc.mtu.edu/boilerplate, date retrieved: March 27, 2018

average, variance, computational time (in seconds), average number of generations until convergence, and success rate of 20 simulations are presented in **Table** 4.2-4.4. The problems are also solved with the initial concept of HGGAs presented in [2]. The results show that Mechanisms A and G, Logic A and G, and alleles concept have better overall performance compared to others. Moreover, based on the overall results presented in **Table** 4.2-4.4, the Egg Holder function seems to be the most difficult problem among the three chosen problems with higher variance and variation in the best solution found. In the Egg Holder function, all the proposed mechanisms except Mechanism B, C, and D could find solutions with lower cost value compared to the initial HGGA concept. In the Schwefel 2.26 and Styblinski-Tang functions, several mechanisms could find solutions with similar or close cost value to the solution of initial HGGA concept. Although the computational time of the initial HGGA concept is lower for both test functions. However in the Schwefel 2.26 function, the success rate of the alleles concept is higher than the initial HGGA concept with the same cost value as the best solution. In the Styblinski-Tang function, the best solution among the tested mechanisms has a cost value of -195.8308 and most of the mechanisms are able to find close solutions to that with high success rate. Mechanisms B, C, D, and Logic B have the highest cost (not desired) and lowest success rate in all the problem. However, this is expected for Logic B since it favors solutions with more hidden genes. In the tested problem however the best solution is to have all the genes active and therefore the performance of Logic B is not good. Regardless, in problems

where the optimal solution has some hidden and some active genes, Logic B might have better performance. The success rate of the Egg Holder function is shown in Figure 4.1 as an example. On each of these boxes, the

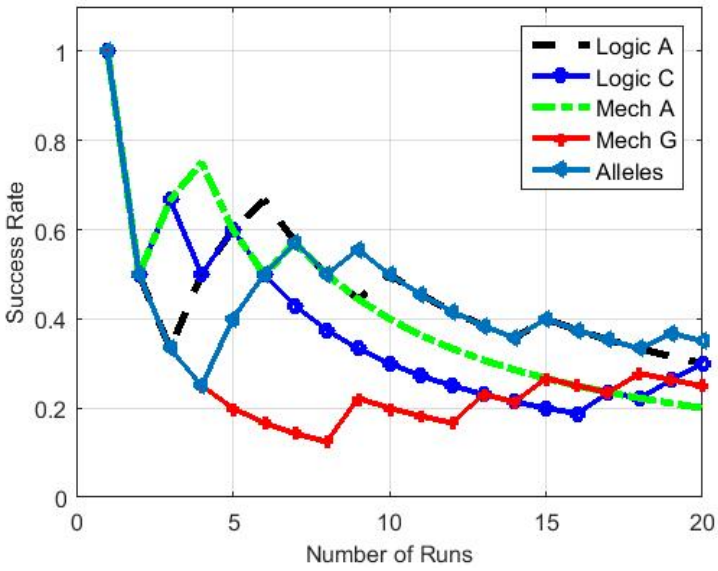


Figure 4.1: Success rate of some mechanisms in Egg Holder function.

By repeating the same numerical experiment, the obtained solution in each experiment is compared to the best obtained solution and a success rate can be updated as the experiment being repeated. A success rate of 0.3 means that if the simulations are repeated ten times, it is expected that three simulations have a cost value of around 95% of the best solution found overall. The box diagram of the mechanisms for the Egg-Holder function is shown in **Figure** 4.2. In this figure, A through H refer to mechanisms A through mechanism H, and LA, LB, and LC refer to logic A, logic B, and logic C, respectively. On each box, the central red mark is the median, the top and bottom edges of the box are the 25th and 75th percentiles, respectively, the

dotted black line is the data considered in the calculations, and the red '+' symbols are the outliers.

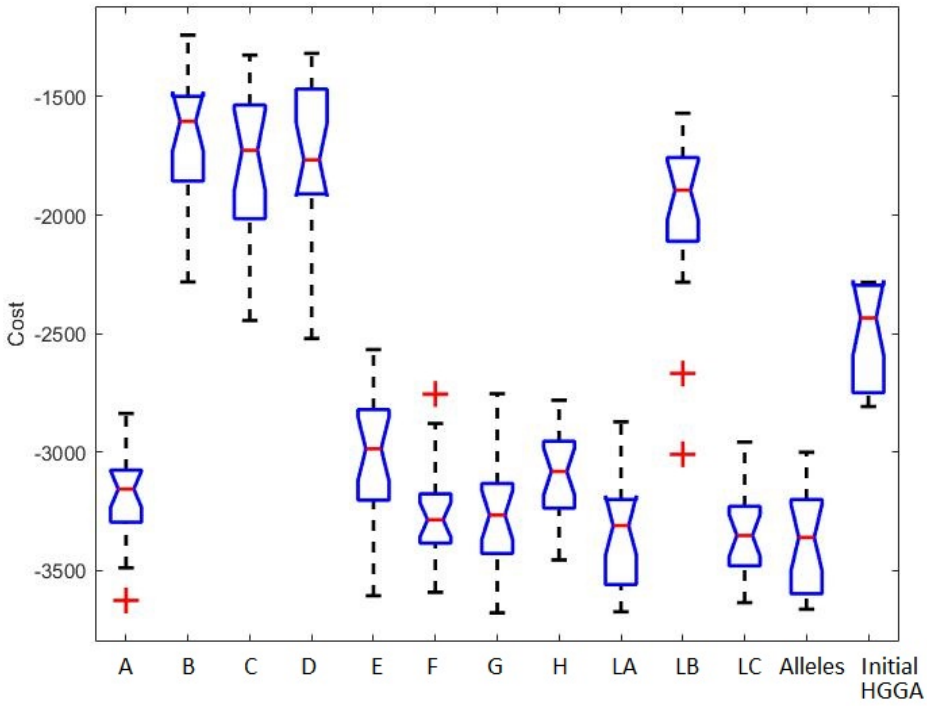


Figure 4.2: Box diagram of all the mechanisms in Egg Holder function.

The results of this section give more insight on the performance of the proposed HGGA mechanisms and can be used as an initial statistical analysis. The mechanisms showed potential for more investigation and hence, in the next section they are tested on space trajectory optimization problems.

Table 4.2
Egg Holder function results

Mechanism	Best	Average	Variance	T_c	N_g	S_R
Logic A	−3674.8391	−3330.7773	55850.2062	191.1277	242.25	30
Logic B	−3010.1431	−1992.4732	127557.2779	176.4492	222	5
Logic C	−3636.4812	−3351.7976	38493.6971	170.9710	215.55	30
Mechanism A	−3627.2132	−3191.2960	39502.0731	121.9006	140.5	20
Mechanism B	−2282.3187	−1665.2146	70489.6712	67.1561	45.1	5
Mechanism C	−2445.1901	−1785.3663	109847.1527	67.6609	47.95	5
Mechanism D	−2521.0772	−1736.6389	85834.8496	63.9779	45.2	5
Mechanism E	−3607.1362	−3014.7806	72811.4486	71.2453	77.65	5
Mechanism F	−3592.4893	−3249.7519	56730.4356	210.2351	285.55	20
Mechanism G	−3679.5732	−3276.8970	56740.0155	185.6655	250.85	25
Mechanism H	−3455.9918	−3097.3507	34409.3814	220.8240	281.8	20
Alleles	−3664.1354	−3377.8405	46136.7272	644.6538	233.35	35
Initial HGGA	−2808.1814	−2496.6042	45111.6674	111.7339	127.8	30

Table 4.3
Schwefel 2.26 function results

Mechanism	Best	Average	Variance	T_c	N_g	S_R
Logic A	−418.9828	−417.6561	28.1062	116.1333	141.35	95
Logic B	−335.1862	−258.0072	963.0126	165.8180	208.55	5
Logic C	−418.9828	−418.8164	0.4639	105.3200	127.8	100
Mechanism A	−418.7245	−410.5598	67.1025	74.5818	78.65	85
Mechanism B	−263.7859	−223.3435	600.5483	74.6767	50.95	10
Mechanism C	−350.4428	−251.4763	1722.6184	80.2498	59	10
Mechanism D	−286.7516	−233.8988	868.0824	76.9453	57.15	10
Mechanism E	−418.2302	−394.9007	517.0262	57.4893	60.7	55
Mechanism F	−418.9828	−417.2749	28.2410	191.4418	256.2	95
Mechanism G	−418.9828	−416.0680	38.4820	140.2377	188.65	95
Mechanism H	−418.8752	−401.3159	310.9379	204.2924	267.95	50
Alleles	−418.9829	−418.9218	0.07207	369.3322	132.4	100
Initial HGGA	−418.9829	−416.6141	53.1574	63.0096	67.05	90

Table 4.4
Styblinski-Tang function results

Mechanism	Best	Average	Variance	T_c	N_g	S_R
Logic A	−195.8307	−195.8281	0.00011	74.4599	84.95	100
Logic B	−195.8307	−155.7955	386.6748	97.8044	115.6	10
Logic C	−195.8308	−195.8302	$1.3061e-06$	71.8116	80.85	100
Mechanism A	−195.8304	−195.7179	0.04677	98.74251	107.65	100
Mechanism B	−176.0926	−158.9485	172.4940	99.1229	70.4	35
Mechanism C	−190.5350	−165.1095	185.9181	90.3370	67.2	15
Mechanism D	−189.2275	−164.6126	151.5187	86.1996	63.35	5
Mechanism E	−195.8097	−188.1418	137.0943	74.4072	83.2	70
Mechanism F	−195.8307	−195.8301	$6.7591e-07$	75.6282	91.9	100
Mechanism G	−195.8308	−195.8304	$2.4139e-07$	75.3374	91.7	100
Mechanism H	−195.8308	−195.1230	9.9912	84.8920	98.5	95
Alleles	−195.8308	−195.8301	$6.4460e-07$	246.5077	83.35	100
Initial HGGA	−195.8308	−195.8308	$1.4025e-15$	49.8379	50.95	100

4.3 Space Trajectory Optimization

The purpose of interplanetary trajectory design is to select different variables such that a spacecraft travels from of celestial body to another with the best objective function. To get to the final destination, the spacecraft can have multiple revolutions around the sun, different flybys around other celestial bodies, and also multiple DSMs in each leg. The number of flybys and DSMs are the variables that describe the topology of the mission and make the problems a VSDS optimization problem. Other variables of this problem include the launch and arrival time, flight direction, time of flight for each leg, pericenter altitude for each flyby, rotation angles, epochs of DSMs, and the DSM vectors (direction and magnitude). These variables can be categorized into two groups of discrete design variables and continuous design variables

(Table 4.5).

Table 4.5
Design variables in an interplanetary trajectory optimization problem

Discrete Variables	Continuous Variables
Number of flybys (m)	Departure date (t_d)
Flyby planets (P)	Arrival date (t_a)
Number of DSMs in each leg (n)	TOF
Flight direction (f_{dir})	Flyby pericenter altitude (h_p)
	Flyby rotation angles (η)
	DSMs epoch (ϵ)
	DSMs magnitudes and directions

It is assumed in this study that the spacecraft operates with impulsive thrust and can have multiple DSMs in each leg. The objective function is to minimize the fuel consumption, which can be divided into departure (launch) impulse, arrival impulse, and DSMs maneuvers.

$$\Delta v_{tot} = ||\Delta V_d|| + ||\Delta V_a|| + \sum_{i=1}^{n} ||\Delta V_{DSM}|| + \sum_{i=1}^{m} ||\Delta V_{ps}|| \tag{4.5}$$

where $||\Delta V_d||$ is the launch impulse, $||\Delta V_a||$ is the arrival impulse, n is the total number of DSM maneuvers, m is the number of powered gravity assist maneuvers, $||\sum_{i=1}^{n} \Delta V_{DSM}||$ is the total costs of DSM maneuvers, and $\sum_{i=1}^{m} ||\Delta V_{ps}||$ is the total post-flyby impulses in the powered gravity assist maneuvers.

When there are no DSMs or flybys for a mission, the trajectory problem becomes a Lambert's problem. Lambert's problem is a two-body boundary value problem that computes the trajectory using initial and final heliocentric position vectors and TOF. The initial and final heliocentric positions of the spacecraft are assumed to be the same as the heliocentric position vector of the home planet and target planet at the initial and final time, respectively. The solution of the Lambert's problem determines the departure and arrival impulses, and hence the transfer orbit.

In the case of an n-impulse trajectory with no flybys (n DSMs in one mission leg), the independent design variables are assumed the departure and arrival time, the ΔV vector of n impulses, and the epoch of the DSMs. Knowing the departure time, the planet heliocentric position vector can be determined (assumed equal to the heliocentric position vector of the spacecraft). Since the epoch of the first DSM and the initial velocity vector are known, the Kepler's equation can be used to propagate the position and velocity vector of the spacecraft at the DSM epoch. The velocity vector of the spacecraft after the DSM can be computed as the summation of the velocity vector of the spacecraft before the DSM and the DSM impulse vector. This procedure is repeated for all the transfer orbits of the trajectory except the last one, where the Lambert's problem is solved. For the last transfer orbit, the arrival time and hence the orbit's TOF are known. The planet's position vector can be determined (equal to the spacecraft position vector at arrival) and therefore, the Lambert's problem can be used. This results in the arrival impulse for capture by the planet.

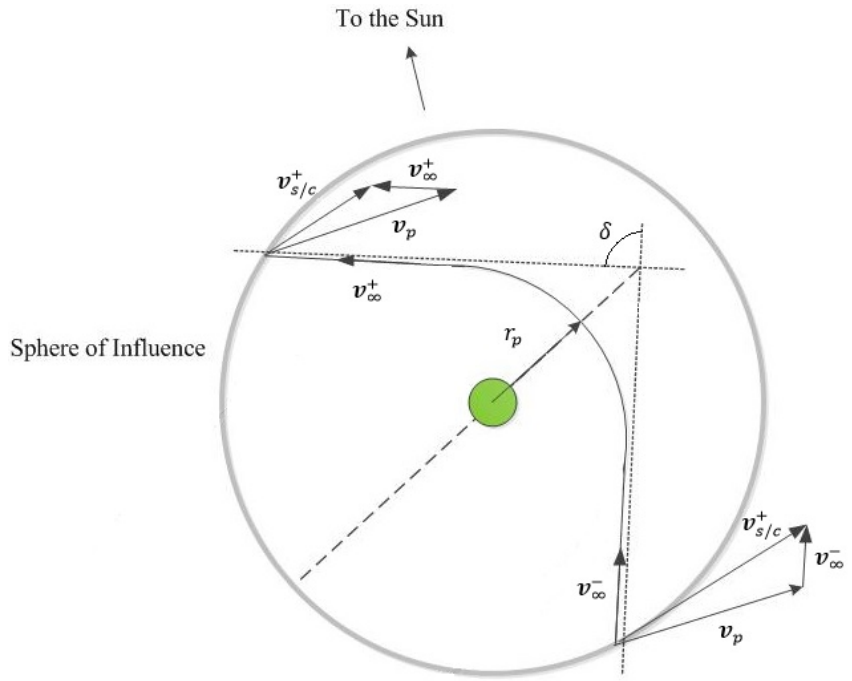


Figure 4.3: Geometry of a non-powered flyby.

The spacecraft can have multiple powered or non-powered gravity-assist maneuvers (flybys). The momentum change in a flyby maneuver can impact the ΔV needed for the spacecraft during the mission. The spacecraft position vector during the flyby is assumed not to change and is equal to the heliocentric position vector of the planet at the flyby instance.

$$\mathbf{r}^- = \mathbf{r}^+ = \mathbf{r}_p \tag{4.6}$$

where $\mathbf{r}^-$ and $\mathbf{r}^+$ are the position vectors of the spacecraft before and after the flyby maneuver and $\mathbf{r}_p$ is the heliocentric position vector of the planet at the flyby instance. The velocity vector of the spacecraft after the flyby maneuver is determined by calculating the magnitude and direction of the velocity for powered and non-powered flybys as follows:

† Non-powered flyby: It is assumed that during the flyby, the linear momentum of the spacecraft changes only due to the gravity field of the planet. Hence, the magnitude of incoming and outgoing relative velocities are the same:

$$|\mathbf{v}_\infty^-| = |\mathbf{v}_\infty^+| = v_\infty \tag{4.7}$$

where $\mathbf{v}_\infty^-$ and $\mathbf{v}_\infty^+$ are the incoming and outgoing relative velocity vectors, respectively and are calculated as:

$$\mathbf{v}_\infty = \mathbf{v}_{S/C} - \mathbf{v}_p \tag{4.8}$$

$\mathbf{v}_{S/C}$ is the spacecraft velocity vector and $\mathbf{v}_p$ is the planet velocity vector (**Figure** 4.3). The direction of the outgoing velocity can be determined by the flyby plane rotation angle δ.

$$sin(\delta/2) = \frac{\mu_p}{\mu_p + r_{per} v_\infty^2} \tag{4.9}$$

where μ_p is the gravitational constant of the planet and r_{per} is the pericenter radius of the flyby which is a design variable. The maximum rotation angle is when the pericenter radius is minimum. If the required rotation angle is greater than the maximum achievable rotation angle, a powered flyby maneuver

is needed. The total spacecraft velocity change in a non-powered flyby is:

$$\Delta v_{npf} = 2v_{\infty} sin(\delta/2) \tag{4.10}$$

† Powered flyby: Higher rotation angles can be gained by applying a small impulse during the flyby [71]. The spacecraft velocity on the periapsis trajectory (v_m) is [72]:

$$v_m = \sqrt{v_{\infty}^2 + 2\mu_p/r_{per}} \tag{4.11}$$

Hence, the required change in velocity for powered flyby is:

$$\Delta v_{pf} = v_m^+ - v_m^- = \sqrt{{v_{\infty}^+}^2 + 2\mu_p/r_{per}} - \sqrt{{v_{\infty}^-}^2 + 2\mu_p/r_{per}} \tag{4.12}$$

The outgoing velocity of the spacecraft in heliocentric inertial frame can be calculated as follows [3]:

$$\mathbf{v}_{\infty}^+ = \mathbf{C}(\mathbf{v}_{\infty}^+)_L \tag{4.13}$$

where $(\mathbf{v}_{\infty}^+)_L$ is the outgoing relative velocity vector expressed in the local frame $\hat{i}\hat{j}\hat{k}$ and $\mathbf{C} = [\hat{i} \quad \hat{j} \quad \hat{k}]$ is the transformation matrix between local frame and inertial frame. As shown in **Figure** 4.4, $(\mathbf{v}_{\infty}^+)_L$ can be calculated as [3]:

$$(\mathbf{v}_{\infty}^+)_L = v_{\infty}[cos(\delta) \quad sin(\delta) \quad 0]^T \tag{4.14}$$

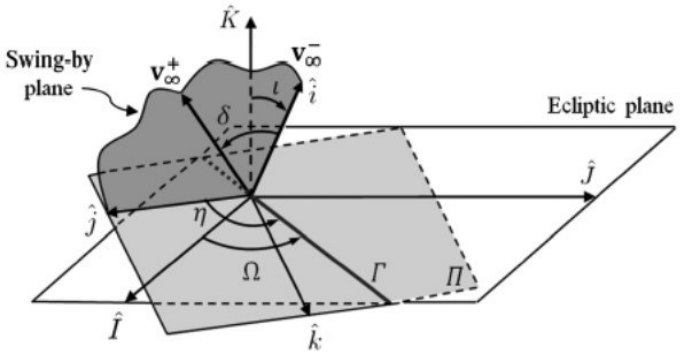


Figure 4.4: The local and inertial frames [3].

The local frame is defined such that $\hat{i}$ is in the direction of the incoming relative velocity and $\hat{j}$ is perpendicular to $\hat{i}$ and is in the plane of the flyby maneuver. Line Γ in **Figure** 4.4 is the intersection of $\hat{j}\hat{k}$ plane (Π plane) and the inertial Ecliptic plane $\hat{I}\hat{J}$. The angle between $\hat{I}$ and Γ is Ω, and the angle between Γ and $\hat{j}$ is η. Also, ι is the inclination of plane Π to the Ecliptic plane. By this nomenclature, the unit directions can be derived as [3]:

$$\hat{i} = \frac{\mathbf{v}_\infty^-}{|\mathbf{v}_\infty^-|} \tag{4.15}$$

$$\hat{j} = \begin{bmatrix} cos(-\Omega) & sin(-\Omega) & 0 \\ -sin(-\Omega) & cos(-\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(-\iota) & sin(-\iota) \\ 0 & -sin(-\iota) & cos(-\iota) \end{bmatrix} \times \begin{bmatrix} cos(-\eta) & sin(-\eta) & 0 \\ -sin(-\eta) & cos(-\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.16}$$

$$\hat{k} = \hat{i} \times \hat{j} \tag{4.17}$$

For the full MGADSM problem with m flybys and n_i DSMs in each leg ($i = 1 \ldots m$), the calculations for each leg is carried out as explained above. Departure and arrival dates and the TOF of each leg (except the last leg) are design variables. The TOF of the last leg can be calculated knowing the total TOF of the mission and the summation of the TOF of the other legs. Assume that there are n_1 DSMs in the first leg. Hence, there are $n_1 + 1$ transfer orbits in that leg. The calculations of the first n_l orbits are similar to the explanations on the n-impulse trajectory. For the last orbit, the velocity vector at the end point is the incoming heliocentric velocity of the flyby. The flyby is assumed non-powered if at least one DSM is in the following leg. Knowing the flyby pericenter altitude and rotation angle (design variables), the outgoing velocity

(the spacecraft initial heliocentric velocity vector for the next leg) can be determined by carrying out the non-powered flyby calculations. This procedure is repeated for all the legs. In case of no DSMs in a leg, the initial flyby of that leg is assumed a powered flyby and the corresponding calculations can be used.

For all the problems, the J2 effect is ignored. Since Lambert's problem can have multiple solutions, the maximum number of revolutions is set to 5 and the best solution from Lambert's problem is selected as the trajectory for the current transfer orbit. The criteria to choose the best Lambert's solution; i.e. choose the number of revolutions, is to select the one that results in the lowest segment cost. This cost can be the post-flyby maneuver, departure impulse, or a DSM impulse.

To illustrate how this problem is a VSDS optimization problem, two sample solutions are shown as chromosomes in **Figure** 4.5. In this example, the hidden genes are shown with gray color. The top part of the figure shows the chromosomes in HGGA, with hidden genes and equal lengths, and the bottom part of the figure shows the equivalent chromosomes with no hidden genes and different lengths. As seen, depending on the number of flybys and DSMs, the length of the solutions can be variable. In the first solution, there is one flyby and one DSM, and in the second solution there are two flybys and two DSMs. Assume that it is required to send a spacecraft to planet Jupiter with the lowest cost (fuel consumption) within certain ranges for launch and arrival dates. The two solutions shown in **Figure** 4.5 can be interpreted as follows:

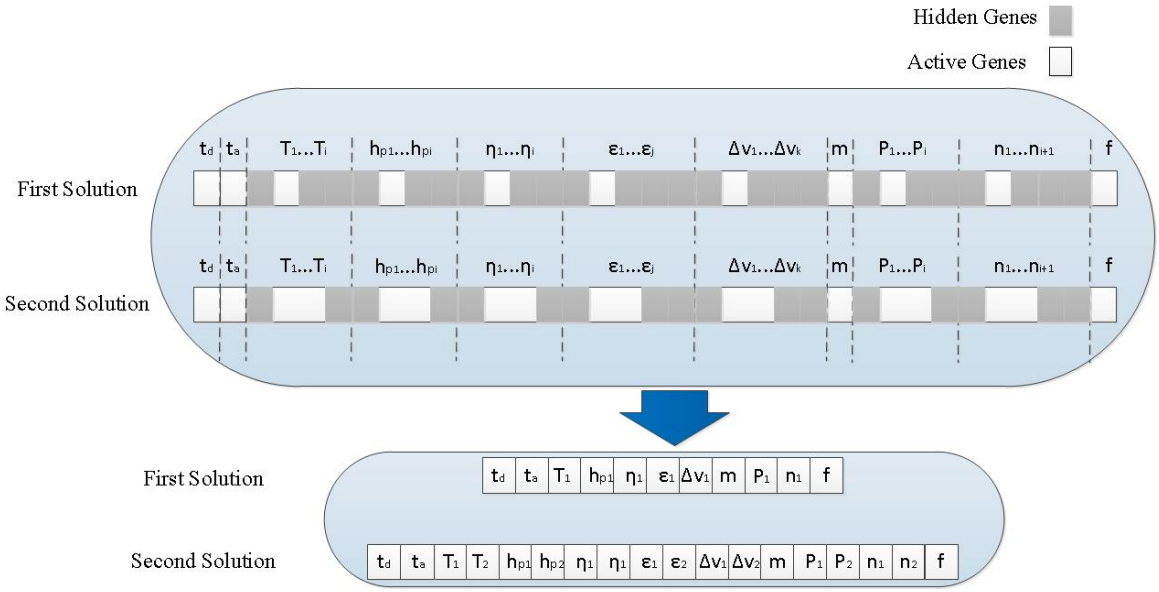


Figure 4.5: An example of two different solutions for an interplanetary trajectory problem in HGGA (Earth to Jupiter), and the equivalent chromosomes with no hidden genes in GA.

1. First Solution: A trajectory with one flyby around Venus (Earth-Venus-Jupiter) and one DSM in the second leg.

2. Second Solution: A trajectory with two flybys around Venus and Earth (Earth-Venus-Earth-Jupiter or EVEJ) and two DSMs in the first and the third legs.

This is a VSDS problem; the proposed tags/Alleles mechanisms can be used to search for the optimal solution and architecture.

In this study, three benchmark problems are investigated: Earth to Mars, Earth to Jupiter, and Earth to Saturn. The best known solutions for these problems can be found in the European Space Agency (ESA) website [3] and also in [3, 73].

[3]European Space Agency, Advanced Concept Team (ESA/ACT),

The HGGA mechanisms are capable to search for the optimal variables including flyby sequences, DSMs' epochs, magnitudes, and directions, launch and arrival dates, and flight direction. To reduce the computational cost, the problems are solved in two phases. In the first phase, the number of design variables is reduced by assuming that there are no DSMs (zero-DSM phase). Reducing the number of design variables in this phase allows for search in larger range of remaining variables. Hence, more planets can be explored for flyby sequence and wider launch, arrival, and flyby dates can be investigated. The second phase is a multi-gravity-assist with DSMs (MGADSM phase) that uses a fixed flyby sequence (obtained in the first step) to optimize the rest of the design variables including the DSMs in the mission. The range of launch, arrival, and flyby dates in the MGADSM phase are selected around the results of the zero-DSM phase. This approach has shown to be computationally efficient [3] compared to a single model where all the variables including DSMs and flyby sequences are optimized together. However, it should be kept in mind that in some missions (such as Messenger), solving the problem in two phases may result in exclusion of fit solutions. For example, some mission trajectories are only fit when there are DSMs and by removing the DSMs, they become unfit and hence get omitted from the zero-DSM phase. Therefore, the feasibility of optimizing the trajectory in two phases should be studied beforehand for each mission.

Each simulation is repeated 100 times for the purpose of statistical analysis on the

http://www.esa.int/gsp/ACT/inf/projects/ gtop/cassini2.html, date retrieved: December 06, 2017

efficiency of the method. For all the problems, the elite count is 20, the crossover probability is 0.95, and the algorithms stops if the average relative change of the best solution over 25 generations is less than or equal to 10^{-6}. The create function for initial population is uniform, the selection function is roulette wheel. Single point crossover and adaptive feasible mutation functions are used for the genes. For the sake of comparison, the lower and upper boundaries of the variables in all problems are compatible with the work done in [3, 73] and the results reported by ESA advanced concept team. Similar to the simulations of the mathematical test functions, Superior, the high-performance computing cluster at the Michigan Technological University, was used in obtaining the results presented in this section.

4.3.1 Earth to Mars Mission Trajectory Optimization

The upper and lower boundaries of the variables are listed in **Table** 4.6. In the mission to Mars, the spacecraft can have up to two flybys around any planet in Solar system and up to two DSMs in each leg. Hence, the chromosome has two genes for the flyby planets in the zero-DSM phase, and each flyby planet can be any one from one (Mercury) to eight (Neptune). Each flyby gene carries the planet identification number. One tag (two tags in the case of using the Alleles concept) is assigned to each flyby gene and if the tag of any of the flybys is one, the corresponding flyby is hidden. For example assume that the values of flybys are three (first flyby is around

the third planet-Earth) and five (second flyby is around the fifth planet-Jupiter). If the tags are $[1, 0]$, the flyby around Earth is hidden and the solution has only one flyby around Jupiter. Similarly, for the MGADSM phase, there can be a maximum two DSMs in each leg. Since the maximum number of flybys is two, the maximum number of legs is three, and hence, the maximum number of DSMs is six. For each DSM, we need to compute the optimal time (T_{DSM}) at which this DSM occurs. A gene and a tag are added for each DSM time T_{DSM}, and hence, there are six gens and six tags for T_{DSMi} $(i = 1 \cdots 6)$ in this mission. Note that if a flyby is hidden, then its leg disappears and all the DSMs in that leg automatically become hidden. Note also that even if a flyby exists, a DSM in its leg can be hidden depending on the value of its own tag. The range for each DSM is set between $[-5, -5, -5]$ km/s and $[5, 5, 5]$ km/s in three directions as shown in **Table** 4.11. Hence, the chromosome will have genes for $6 \times 3 = 18$ scalar components of the DSMs. Note that these 18 genes are classified in groups of three genes; hence if one DSM is hidden then its three genes get hidden together. In the zero-DSM phase, the launch date range is 01 June 2004 to 01 July 2004 and the arrival date range is 01 April 2005 to 01 July 2005. The TOF for each leg is between 40 and 300 days except the last one. The duration of the last leg is determined by the launch and arrival dates and the TOF of the other legs. There is a gene for each TOF in the mission. Hence, we have three genes for the TOFs in this Mars mission. Note that there are no tags associated with the TOF genes since the state of each gene (hidden or active) is determined based on the flyby

tags. If a flyby exists then there is an active gene for a TOF associated with it. Two genes for the two flyby altitudes h_p and two genes for the two flyby plane angles η are added. Similar to the TOF variables, no tags are needed for the h_p and η genes. There are also six genes for the departure impulse, flight direction, the arrival date and the departure date.

Table 4.6
Lower and upper bounds of Earth-Mars problem

Design Variable	Lower Bound	Upper Bound
Flyby 1 planet	1 ($Mercury$)	8 ($Neptune$)
Flyby 2 planet	1	8
DSM$_i$ (km/s), $i = 1 \cdots 6$	$[-5, -5, -5]$	$[5, 5, 5]$
Flight Direction	Posigrade	Retrograde
Departure Date (t_0)	01 Jun.2004	01 Jul.2004
Arrival Date (t_f)	01 Apr.2005	01 Jul.2005
TOF (days)	$[40, 40]$	$[300, 300]$
Flyby normalized pericenter altitude (h_p)	$[0.1, 0.1, 0.1]$	$[10, 10, 10]$
Flyby plane rotation angle (η) (rad)	$[0, 0, 0]$	$[2\pi, 2\pi, 2\pi]$
Epoch of DSMs ($\epsilon_i, i = 1 \cdots 6$)	0.1	0.9

The problem is solved in two phase of zero-DSM and MGADSM models. For the zero-DSM phase, the number of population is 200 and the number of generations is 100. The results of zero-DSM phase are presented in **Table** 4.7. All the mechanisms found Earth-Venus-Mars (EVM) flyby sequence as the optimal solution. For the MGADSM phase, the number of population is 300 and the number of generations is 200. The results of this phase are presented in **Table** 4.8. For both phases, the best solution found (Best-km/s), average of the 100 simulation results (Average-km/s), the variance of the 100 simulation results (Variance-km/s), the computational time for 100 simulations (hr), the average generations until convergence (N_g), and the success

rate ($S_R\%$) are presented. As shown, mechanism H results in the lowest cost value. The zero-DSM and MGADSM trajectories for mechanism H and alleles concept are shown in **Figure** 4.6 and 4.7 and the mission details for these two mechanisms are presented in **Table** 4.9 and 4.10. Based on the information in these tables, both of these algorithms are able to reduce departure and arrival impulses in the MGADSM phase by adding one DSM in the first leg and tuning the event times.

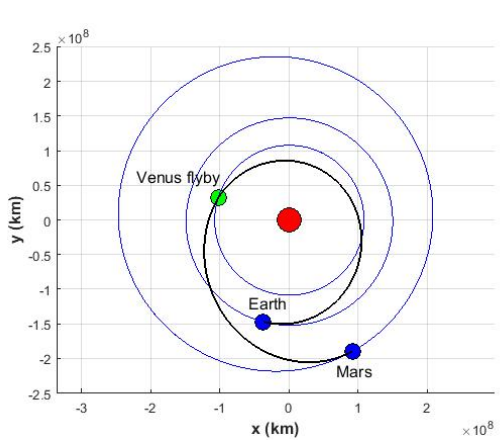
Table 4.7
Results of Earth-Mars problem in zero-DSM phase.

Mechanism	Best	Average	Variance	T_c	N_g	S_R
Mechanism A	10.7800	10.9732	0.0700	7.1207	85.9300	93
Mechanism B	10.7792	11.1489	0.1892	15.1288	69.2600	82
Mechanism C	10.7805	11.1097	0.0921	15.3558	73.5400	88
Mechanism D	10.7890	11.1339	0.2156	14.6486	69.1700	92
Mechanism E	10.7854	10.9843	0.0500	6.7078	81.7100	90
Mechanism F	10.7819	10.9176	0.0241	7.4285	86.2300	98
Mechanism G	10.7797	10.9378	0.0326	7.1350	86.2100	95
Mechanism H	10.7788	11.0159	0.0819	7.7926	86.59	89
Logic A	10.7802	10.9386	0.0425	7.0009	82.8000	94
Logic B	10.7881	12.7345	11.2591	6.8406	83.1200	27
Logic C	10.7798	10.9048	0.0309	6.6724	78.1600	96
Alleles	10.7801	10.9006	0.0293	49.0959	87.16	97

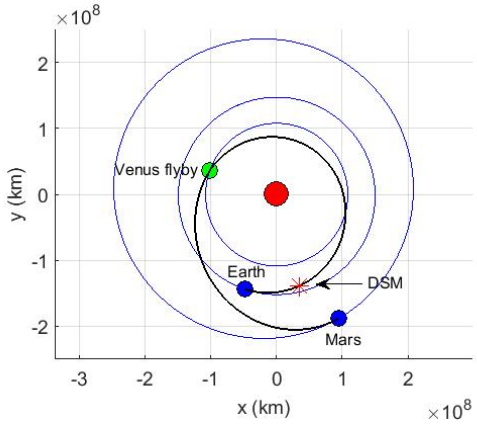
The box diagram of the mechanisms for the MGADSM phase of Earth to Mars problem is shown in **Figure** 4.8. In this figure, A through H refer to mechanisms A through mechanism H, and LA, LB, and LC refer to logic A, logic B, and logic C, respectively. On each box, the central red mark is the median, the top and bottom edges of the box are the 25th and 75th percentiles, respectively, the dotted black line is the data considered in the calculations, and the red '+' symbols are the outliers. As shown, mechanisms A, E, F, G, H, logic A, and alleles have better performance

Table 4.8
Results of Earth-Mars problem in MGADSM phase.

Mechanism	Best	Average	Variance	T_c	N_g	S_R
Mechanism A	10.7778	10.9408	0.0134	30.3534	151.58	100
Mechanism B	10.8122	11.7328	0.3121	19.2612	58.56	37
Mechanism C	10.9309	12.0713	0.4475	14.2896	48.11	20
Mechanism D	10.8652	12.0938	0.3393	15.2857	47.11	34
Mechanism E	10.7626	10.9102	0.0123	17.1103	138.36	99
Mechanism F	10.7622	10.9130	0.0137	20.4235	155.17	99
Mechanism G	10.7569	10.8976	0.0141	19.2714	158.55	99
Mechanism H	10.7461	10.9217	0.0179	21.4197	162.32	98
Logic A	10.7857	10.9354	0.01944	20.7456	172.63	98
Logic B	10.7940	11.2717	0.1602	20.6973	166.58	69
Logic C	10.8481	12.1110	0.4971	5.6413	50.56	25
Alleles	10.7121	10.9178	0.0166	110.2948	132.4300	99



(a) Zero-DSM trajectory.



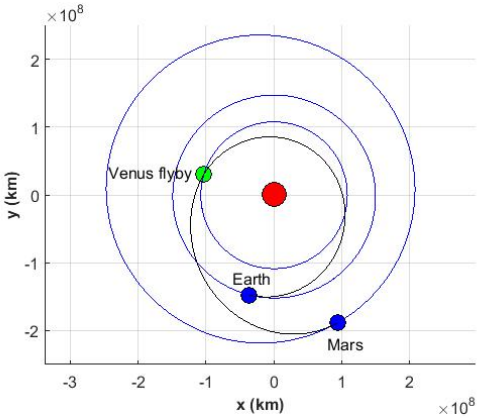
(b) MGADSM trajectory has one DSM in the first leg.

Figure 4.6: Zero-DSM and MGADSM trajectories for Earth to Mars mission using mechanism H.

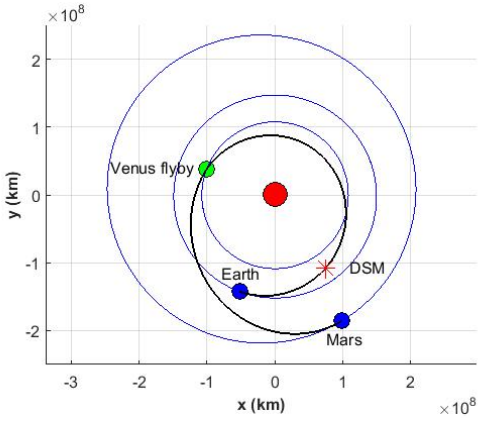
compared to the rest of the mechanisms.

Table 4.9
Solution of Earth to Mars mission (EVM) using mechanism H

Mission parameter	Zero-DSM model	MGADSM model
Departure Date	$05-Jun-2004, 16:13:56$	$01-Jun-2004, 06:00:36$
Departure Impulse (km/s)	4.6121	3.9777
DSM date	–	$08-Jul-2004, 15:06:54$
DSM impulse (km/s)	–	$\|[-0.6080, -0.2763, -0.1587]\| = 0.6864$
Venus flyby date	$20-Nov-2004, 10:59:40$	$18-Nov-2004, 20:04:05$
Pericenter altitude (km)	8051.0917	8107.4551
Arrival date	$13-May-2005, 16:11:09$	$14-May-2005, 21:41:46$
Arrival impulse (km/s)	6.1667	6.082
TOF (days)	$167.7818, 174.2163$	$170.5858, 177.0678$
Mission duration (days)	341.9981	347.6536
Mission cost (km/s)	10.7788	10.7461



(a) Zero-DSM trajectory.



(b) MGADSM trajectory has one DSM in the first leg.

Figure 4.7: Zero-DSM and MGADSM trajectories for Earth to Mars mission using alleles concept.

Table 4.10
Solution of Earth to Mars mission (EVM) using alleles concept

Mission parameter	Zero-DSM model	MGADSM model
Departure Date	$05-Jun-2004, 23:19:34$	$30-May-2004, 17:27:03$
Departure Impulse (km/s)	4.6167	4.0871
DSM date	–	$28-Jul-2004, 18:57:12$
DSM impulse (km/s)	–	$\|[-0.4747, -0.3205, -0.1426]\| = 0.5903$
Venus flyby date	$20-Nov-2004, 14:37:28$	$18-Nov-2004, 03:17:31$
Pericenter altitude (km)	8042.5595	8015.4727
Arrival date	$14-May-2005, 16:42:04$	$17-May-2005, 06:22:58$
Arrival impulse (km/s)	6.1631	6.0347
TOF (days)	$167.6374, 175.0865$	$171.4101, 180.1288$
Mission duration (days)	342.724	351.5388
Mission cost (km/s)	10.7461	10.7121

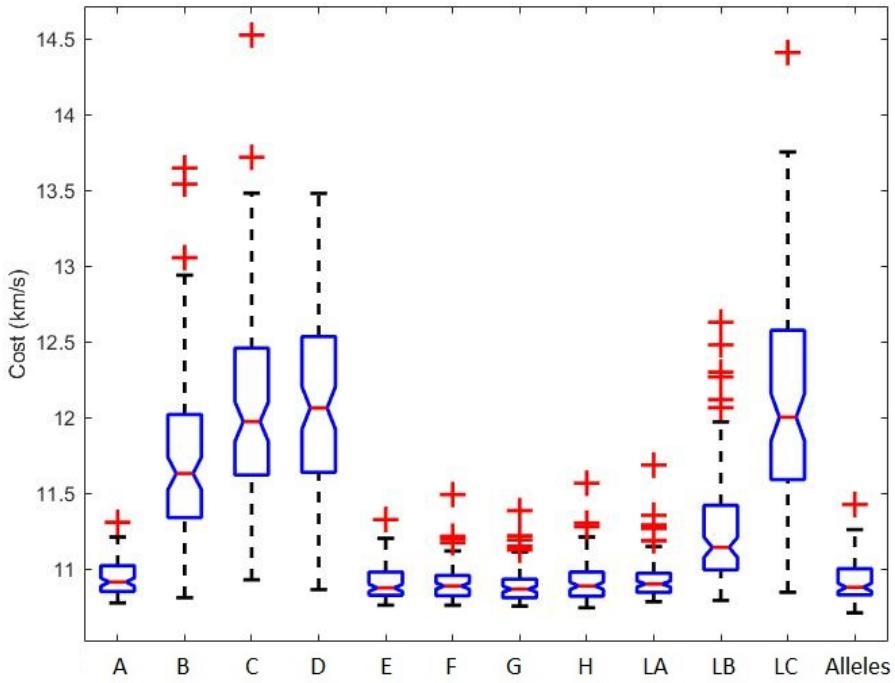


Figure 4.8: Box diagram of all the mechanisms in Earth to Mars problem (MGADSM phase).

4.3.2 Earth to Jupiter Mission Trajectory Optimization

The variable boundaries are listed in **Table** 4.11. The spacecraft can flyby around up to two planets in the solar system and there can be up to two DSMs in each leg. For the zero-DSM phase, the launch date can be between 01 September 2016 and 30 September 2016 and the arrival date can be between 01 September 2021 and 31 December 2021. The TOF of each leg (except the last leg) can be between 80 and 800 days. In the MGADSM phase, there can be up to two DSMs in each leg (total possible number of DSMs is six for maximum three legs) and the range for each DSM is $[-5, -5, -5]$ km/s to $[5, 5, 5]$ km/s.

Table 4.11
Lower and upper bounds of Earth-Jupiter problem

Design Variable	Lower Bound	Upper Bound
Flyby 1 planet	1 (*Mercury*)	8 (*Neptune*)
Flyby 2 planet	1	8
DSM_i (km/s), $i = 1 \cdots 6$	$[-5, -5, -5]$	$[5, 5, 5]$
Flight Direction	Posigrade	Retrograde
Departure Date (t_0)	01 Sep.2016	30 Sep.2016
Arrival Date (t_f)	01 Sep.2021	31 Dec.2021
TOF (days)	$[80, 80]$	$[800, 800]$
Flyby normalized pericenter altitude (h_p)	$[0.1, 0.1, 0.1]$	$[10, 10, 10]$
Flyby plane rotation angle (η) (rad)	$[0, 0, 0]$	$[2\pi, 2\pi, 2\pi]$
Epoch of DSMs ($\epsilon_i, i = 1 \cdots 6$)	0.1	0.9

The variables that have tags and the implementation of tags are similar to the description provided in **Section** 4.3.1 for Earth to Mars mission. For the zero-DSM model, the population size is set to 400 and the number of generations is 300. This problem

is solved using different mechanisms and the best cost values, computational time, variance, and average cost of each mechanism in the zero-DSM model are reported in **Table** 4.12.

Table 4.12
Results of Earth-Jupiter problem in zero-DSM model

Mechanism	Best	Average	Variance	T_c	N_g	S_R
Mechanism A	10.1293	15.0380	11.1995	35.7936	186.6000	11
Mechanism B	16.4544	22.8396	3.5713	29.5913	72.0200	1
Mechanism C	11.6747	22.4008	8.8540	22.0437	49.1200	2
Mechanism D	12.2634	22.2984	8.6277	25.8117	57.4500	2
Mechanism E	10.1643	13.6744	1.3808	14.2629	78.2200	7
Mechanism F	10.1294	15.4224	11.0956	29.9687	150.2400	12
Mechanism G	10.1283	15.5772	10.5792	32.0933	159.3100	11
Mechanism H	10.1214	16.0900	13.1852	32.9162	164.5500	12
Logic A	10.2654	14.8729	10.0933	38.4243	162.3200	23
Logic B	17.1731	23.1896	5.0628	17.8781	107.8600	1
Logic C	10.1635	13.4421	10.3030	44.0043	180.47	32
Alleles	10.1448	13.7523	9.8504	69.7166	97.12	11

All the mechanisms could find the known optimal flyby sequence which is Earth-Venus-Earth-Jupiter (EVEJ) in their zero-DSM model. The flyby sequences is set in the MGADSM model and the range of time of launch, arrival, and flybys are set around 10 days of the results of the zero-DSM model. The cost of final MGADSM mission scenarios are presented in **Table** 4.13 for a population size of 600 and 200 generations. Note that the computational time is presented for 100 simulations.

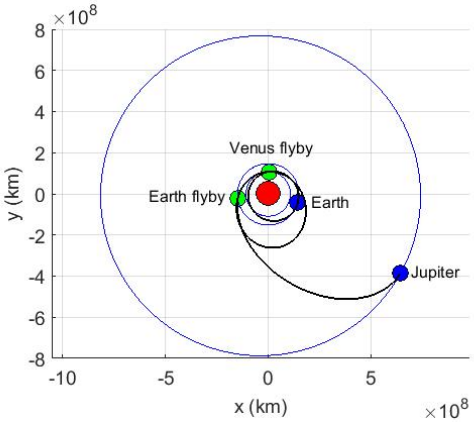
Mechanism E has the lowest cost of 10.1179 km/s with one DSM in the first leg. Mechanism H has the second lowest cost of 10.1214 km/s. The zero-DSM and MGADSM trajectories using mechanisms E and H are shown in **Figures** 4.9 and 4.10. The

Table 4.13
Results of Earth-Jupiter problem in MGADSM model

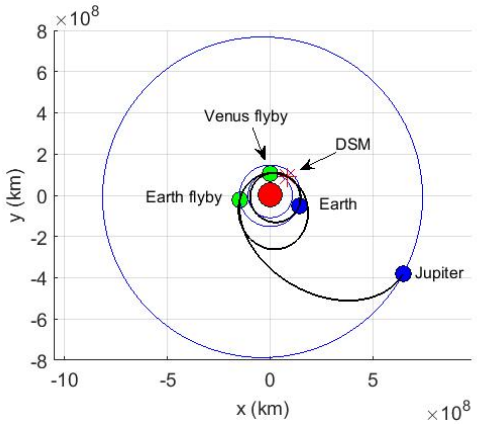
Mechanism	Best	Average	Variance	T_c	N_g	S_R
Mechanism A	10.1355	10.5788	0.1151	47.6756	175.61	69
Mechanism B	13.9969	14.7439	0.2144	57.0613	73.03	49
Mechanism C	10.8836	11.7026	0.2467	62.0653	73	29
Mechanism D	10.8786	12.0616	0.2227	60.7960	87.75	8
Mechanism E	10.1180	10.2134	0.0105	67.4248	163.94	94
Mechanism F	10.1296	10.3687	0.0501	67.4763	184.96	88
Mechanism G	10.1236	10.3560	0.0372	68.0114	187.16	91
Mechanism H	10.1226	10.3430	0.0484	69.3444	183.39	93
Logic A	10.2139	10.4654	0.0369	59.9637	186.49	91
Logic B	14.0996	15.6615	0.2108	13.3010	97.18	3
Logic C	11.2685	15.5756	6.2181	16.6324	75.11	4
Alleles	10.1437	10.3193	0.0148	360.6664	158.81	100

detailed mission scenario are presented in **Table** 4.14 and 4.15.

As a demonstration for how the tags evolve over subsequent generations, consider this Earth to Jupiter problem solved using Logic C (MGADSM phase). The population size is 300 and the number of generations is 100. Six tags are examined. Figure 4.11



(a) Zero-DSM trajectory.

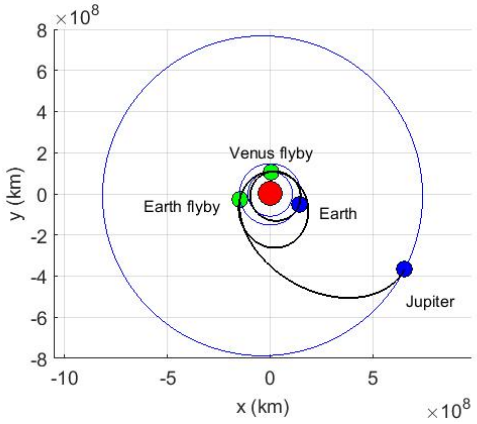


(b) MGADSM trajectory has one DSM in the first leg.

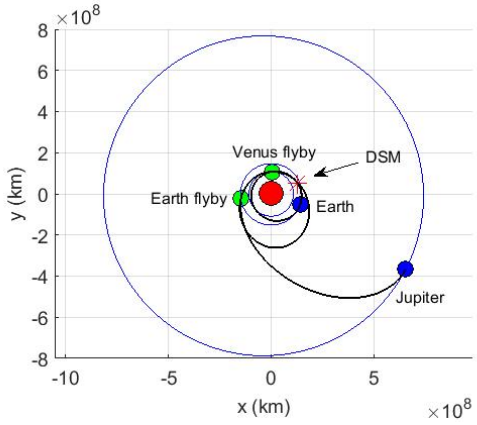
Figure 4.9: Zero-DSM and MGADSM trajectories for Earth to Jupiter mission using mechanism E.

Table 4.14
Solution of Earth to Jupiter (EVEJ) mission using mechanism E

Mission parameter	Zero-DSM model	MGADSM model
Departure Date	$05-Sep-2016, 04:43:08$	$01-Sep-2016, 16:36:49$
Departure Impulse (km/s)	3.5147	3.3834
DSM date	–	$09-Nov-2016, 05:05:37$
DSM impulse (km/s)	–	$\|[0.0705, -0.0653, -0.0019]\| = 0.0962$
Venus flyby date	$05-Sep-2017, 09:12:37$	$06-Sep-2017, 18:20:24$
Pericenter altitude (km)	1261.1915	909.9072
Earth flyby date	$29-Mar-2019, 06:14:18$	$29-Mar-2019, 05:55:57$
Post-flyby impulse (km/s)	0.4453	0.4412
Pericenter altitude (km)	637.7999	637.8000
Arrival date	$08-Sep-2021, 22:50:57$	$13-Sep-2021, 10:16:34$
Arrival impulse (km/s)	6.2043	6.1972
TOF (days)	$365.1871, 569.8762, 894.6921$	$370.07193, 568.483, 899.181$
Mission duration (days)	1829.7554	1837.7359
Mission cost (km/s)	10.1643	10.1180



(a) Zero-DSM trajectory.



(b) MGADSM trajectory has one DSM in the first leg.

Figure 4.10: Zero-DSM and MGADSM trajectories for Earth to Jupiter mission using mechanism H.

shows the number of times each tag has a value of '1' in each generation. For example, tag 6 takes a value of '1' in all the population members in generations 55 and above. In the 30^{th} generation, for instance, tag 6 takes a value of '1' in only 40 chromosomes and takes a value of '0' in the other 260 chromosomes. The other 5 tags converge to a value of '0' in the last population in all the chromosomes.

Table 4.15
Solution of Earth to Jupiter (EVEJ) mission using mechanism H

Mission parameter	Zero-DSM model	MGADSM model
Departure Date	$01-Sep-2016, 20:07:38$	$01-Sep-2016, 10:12:43$
Departure Impulse (km/s)	3.4854	3.4807
DSM date	–	$17-Oct-2016, 23:11:31$
DSM impulse (km/s)	–	$\|[0.00904, -0.0216, 0.0020]\| = 0.02348$
Venus flyby date	$05-Sep-2017, 12:08:44$	$05-Sep-2017, 13:40:00$
Pericenter altitude (km)	1318.4348	1268.1977
Earth flyby date	$30-Mar-2019, 05:04:19$	$29-Mar-2019, 20:47:51$
Post-flyby impulse (km/s)	0.4407	0.4423
Pericenter altitude (km)	637.7999	637.8000
Arrival date	$29-Sep-2021, 04:41:57$	$27-Sep-2021, 08:06:04$
Arrival impulse (km/s)	6.1961	6.1889
TOF (days)	$368.6674, 570.7053, 913.9845$	$369.14395, 570.2971, 912.471$
Mission duration (days)	1853.3572	1851.912
Mission cost (km/s)	10.1214	10.1226

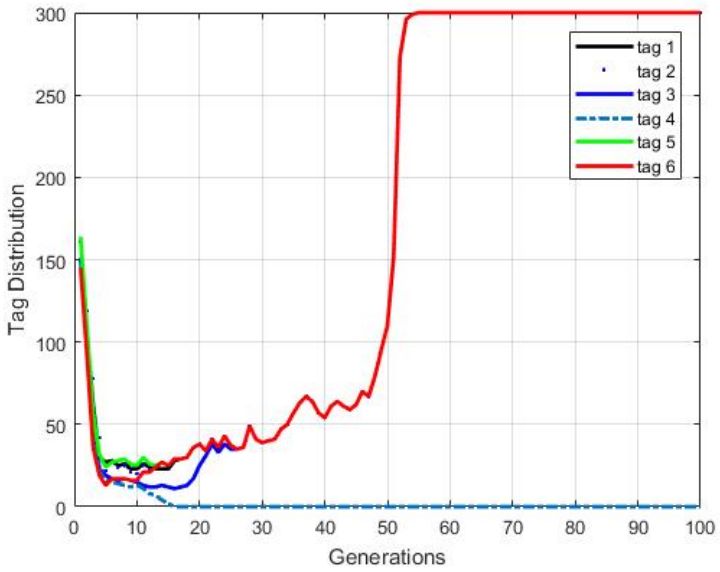


Figure 4.11: Evolution of tags using Logic C in the Earth to Jupiter problem

The box diagram of the mechanisms for the MGADSM phase of Earth to Jupiter problem is shown in **Figure** 4.12. In this figure, A through H refer to mechanisms A through mechanism H, and LA, LB, and LC refer to logic A, logic B, and logic C, respectively. On each box, the central red mark is the median, the top and bottom

edges of the box are the 25th and 75th percentiles, respectively, the dotted black line is the data considered in the calculations, and the red '+' symbols are the outliers. Similar to Earth to Mars simulation results, mechanisms A, E, F, G, H, logic A, and alleles have better performance compared to the rest of the mechanisms.

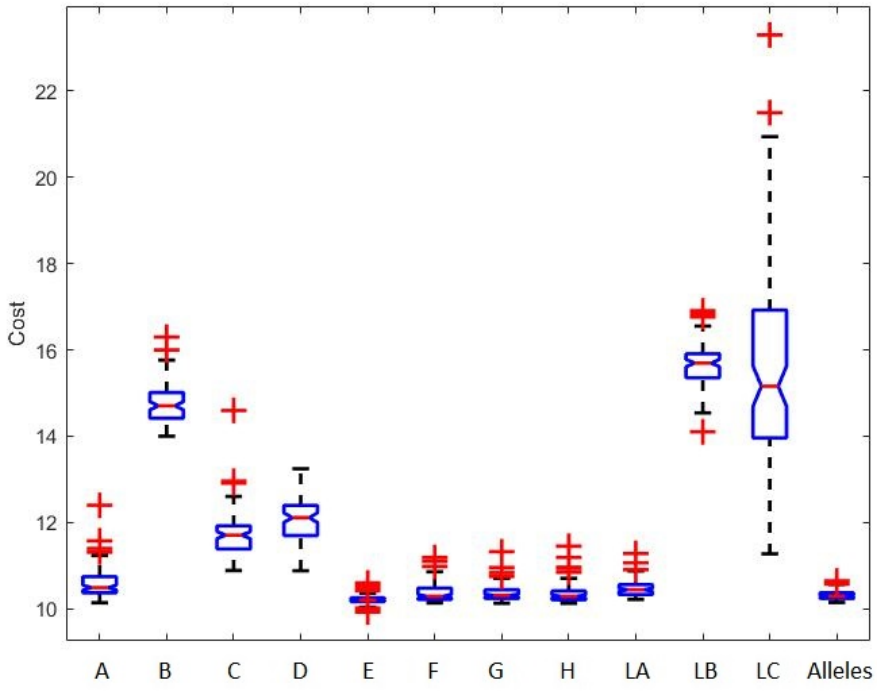


Figure 4.12: Box diagram of all the mechanisms in Earth to Jupiter problem (MGADSM phase).

4.3.3 Earth to Saturn (Cassini 2) Mission Trajectory Optimization

A more complicated trajectory is the Cassini mission that was designed by NASA, European Space Agency, and Italian Space Agency to discover the planet Saturn. The mission consists of a satellite that orbits Saturn and a lander for its moon Titan [74]. We only consider the first stage of the mission to design the trajectory from Earth to rendez-vous with Saturn. High number of flybys and wide ranges for the design variables make this problem challenging for optimizer tools. Here, a launch window of 30 days is selected for the mission for the sake of comparison with the reported results in the literature [3, 75]. The upper and lower boundaries of design variables are shown in **Table** 4.16. For a fair comparison, these ranges are selected in accordance to the literature. There can be up to four flybys (five legs) and up to one DSM in each leg. Hence, the maximum possible number of DSMs in the mission is five.

The goal here is to optimize the trajectory to Cassini as a VSDS problem with unknown number of flybys and DSMs. However, the initial simulations show that the algorithms converge to local solutions with higher cost value than reported in the literature. A niching method is used to help the optimization algorithms explore more

Table 4.16
Lower and upper bounds of Earth-Saturn problem

Design Variable	Lower Bound	Upper Bound
Flyby planet identification (P_i, $i = 1 \dots 4$)	$2\,(Venus)$	$5\,(Jupiter)$
DSM_i (km/s), $i = 1 \cdots 5$	$[-5, -5, -5]$	$[5, 5, 5]$
Flight Direction	Posigrade	Retrograde
Departure Date (t_0)	01 Nov.1997	01 Sep.1997
Arrival Date (t_f)	01 Jan.2007	30 Jun.2007
TOF (days)	$[100, 100, 30, 400]$	$[400, 500, 300, 1600]$
Flyby normalized pericenter altitude (h_p)	$[0.05, 0.05, 0.15, 0.7]$	$[5, 5, 5.5, 290]$
Flyby plane rotation angle (η) (rad)	$[0, 0, 0, 0]$	$[2\pi, 2\pi, 2\pi, 2\pi]$
Epoch of DSMs ($\epsilon_i, i = 1 \cdots 6$)	0.1	0.9

of the design space [3, 35, 57]. In this niching method, every 10 generations, the current best solutions is saved (as a vector). In each generation, the solutions that have similar flyby sequence or the solutions that are close to the saved best solution vectors are given high cost. Moreover, every five generations a random solution is inserted in place of an elite solution. For example if in the 10th generation the best solution is in the form of $\vec{x^*_{g10}} = (x_1, \dots, x_{lm})$ where lm is the total number of variables, then in genrations 11th to 20th, the solutions with $|\vec{x} - \vec{x^*_{g10}}| < 1$ are given high cost. In the 20th generation, another $\vec{x^*_{g20}}$ is saved and from generations 21th to 30th, the solutions with similar flyby sequence to the flyby sequence of $\vec{x^*_{g10}}$ or $\vec{x^*_{g20}}$, or the solutions that are close to these two points in $\mathbb{R}^{lm}$ are given high cost. These simulations are carried out ten times with population size of 400, generation number of 200, elite count of 40, and stall generation limit of 200. After the simulations are done, a vector of niched flyby sequences are available for each mechanism. The simulations are repeated for each flyby sequence and no DSMs (fixed-sized design space problem) and

Table 4.17
Success rates of Earth-Saturn problem in zero-DSM model

Mechanism	S_R
Mechanism A	0
Mechanism B	0
Mechanism C	0
Mechanism D	0
Mechanism E	10
Mechanism F	20
Mechanism G	20
Mechanism H	50
Logic A	50
Logic B	0
Logic C	70
Alleles	30

their cost are compared. The flyby sequence of Earth-Venus-Venus-Earth-Jupiter-Saturn (EVEEJS) has the lowest cost of 10.7960 km/s compared to other niched solutions. Mechanisms E, F, G, H, logics A, and C, and alleles concept are able to find the optimal flyby sequence of Earth-Venus-Venus-Earth-Jupiter-Saturn. The success rate of all these mechanisms are presented in **Table** 4.17.

For the MGADSM phase, only the mechanisms that were able to find the optimal sequence are investigated. The launch and arrival dates are set to a range between the results of the zero-DSM phase. The launch date can be between 18th and 24th of November 1997 and the arrival date is set to be between 21st and 27th of May 2007. The results are summarized in **Table** 4.18 for 30 identical simulations. The population size is 600 and number of generations is 1000. The crossover fraction is 95% and the stall generation limit is 1000.

Table 4.18
Results of Earth-Saturn problem in MGADSM model

Mechanism	Best	Average	Variance	T_c	S_R
Mechanism E	9.8693	10.4591	0.0823	40.3351	20
Mechanism F	9.8444	10.4924	0.0634	112.9543	13.33
Mechanism G	10.5544	10.5899	$9.1224e-4$	42.2296	100
Mechanism H	9.8352	10.5233	0.0708	112.9883	13.33
Logic A	10.0789	10.5631	0.0160	112.2977	41.37
Logic C	10.2682	11.2405	1.2425	106.8431	66.67
Alleles	10.8188	11.6096	0.2092	180.2406	40

The optimized mission trajectory using mechanism H is is shown in **Figure** 4.13 and the mission details are presented in **Table** 4.19.

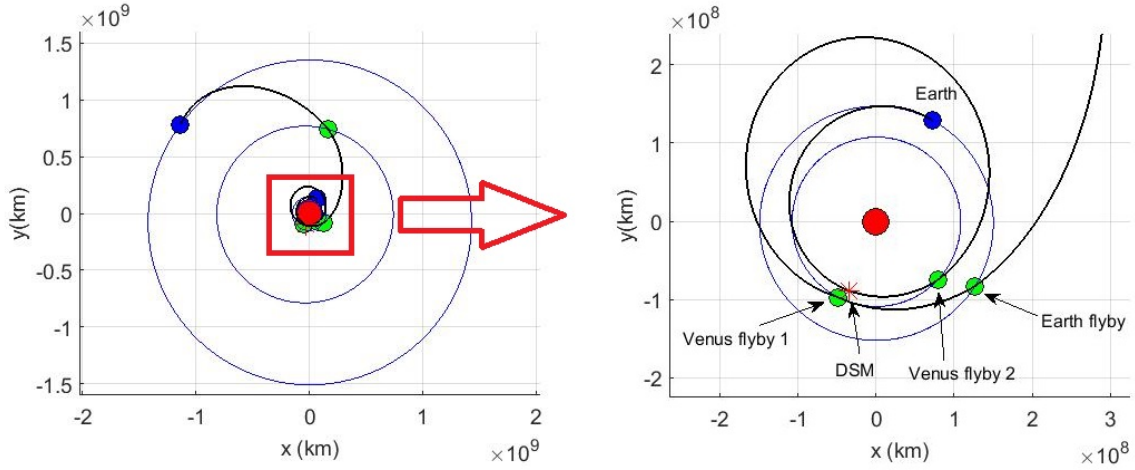


Figure 4.13: MGADSM trajectory for Earth to Saturn mission using mechanism H.

4.4 Discussion

The Global Trajectory Optimization Problems (GTOP) database consists of a wide variety of problems to asteroids and different planets, including Saturn and Mercury.

Table 4.19
Solution of Earth to Saturn (EVVEJS) mission using mechanism H

Mission parameter	MGADSM model
Departure Date	$22 - Nov - 1997, 15 : 40 : 15$
Departure Impulse (km/s)	4.2582
DSM date	$17 - Apr - 1998, 10 : 20 : 24$
DSM impulse (km/s)	$\|[1.0944, -0.39532, -0.46847]\| = 1.2543$
Venus flyby date	$21 - May - 1998, 00 : 44 : 41$
Pericenter altitude (km)	11653.4119
Venus flyby date	$28 - Jun - 1999, 02 : 52 : 55$
Post-flyby impulse (km/s)	0.09639
Pericenter altitude (km)	340.9872
Earth flyby date	$19 - Aug - 1999, 21 : 12 : 37$
Pericenter altitude (km)	2086.3034
Jupiter flyby date	$04 - Apr - 2001, 10 : 26 : 32$
Pericenter altitude (km)	4791311.7441
Arrival date	$25 - May - 2007, 16 : 33 : 02$
Arrival impulse (km/s)	4.2352
TOF (days)	$179.3781, 403.089, 52.7637, 593.5513, 2242.2545$
Mission duration (days)	3471.0367
Mission cost (km/s)	9.8441

The results presented in this chapter show that the proposed mechanisms are capable of finding the optimal architecture of the mission (optimal flyby sequence as well as optimal number of DSMs).

The three investigated problems or variations of them have been studied in different researches. For the sake of comparison, only the methods that assume an impulsive thrust are considered here. The Earth to Mars trajectory optimization problem has been previously solved using extended primer vector theory [75]. In that solution, to implement the primer vector method, the departure and arrival dates were assumed fixed with a mission duration of 340 days. Moreover, it was assumed that the flyby

around Venus is known to occur at 165 days from the departure. The results of that work was a mission with one DSM in the first leg with a total cost of 10.786 km/s. Gad et al.[3] implemented the initial HGGA concept (feasibility criteria) to optimize the mission with unknown number of flybys and DSMs. The result was a mission with one flyby around Venus and one DSM in the first leg of the mission with a cost of 10.728 km/s. In another work using Dynamic-size multiple population GA, the problem is solved with unknown number of flybys and DSMs and the solution obtained is the EVM sequence with one DSM in each leg of the mission. The total cost of this algorithm is 10.7 km/s. The structured-chromosome evolutionary algorithms solution to this problem has a cost of 10.7788 km/s by optimizing the flyby sequence and DSM structure [24]. All the mechanisms proposed in this dissertation are able to find the EVM flyby sequence without any prior knowledge of the mission. The lowest cost of this mission is obtained using alleles concept with one DSM in the first leg and total cost of 10.7121 km/s with a success rate of 99%.

The success rate of all the proposed mechanisms in Earth to Mars problem (MGADSM model) are shown is Figure4.14. As shown in this figure, the success rate curves of all the mechanisms show a convergent behavior for 100 number of identical simulation runs. Seven mechanisms including mechanisms A, E, F, G, H, logic A, and alleles concept have success rates of higher than 90% with low objective function values (low mission cost). The other five mechanisms have lower success rate while their cost values are relatively higher.

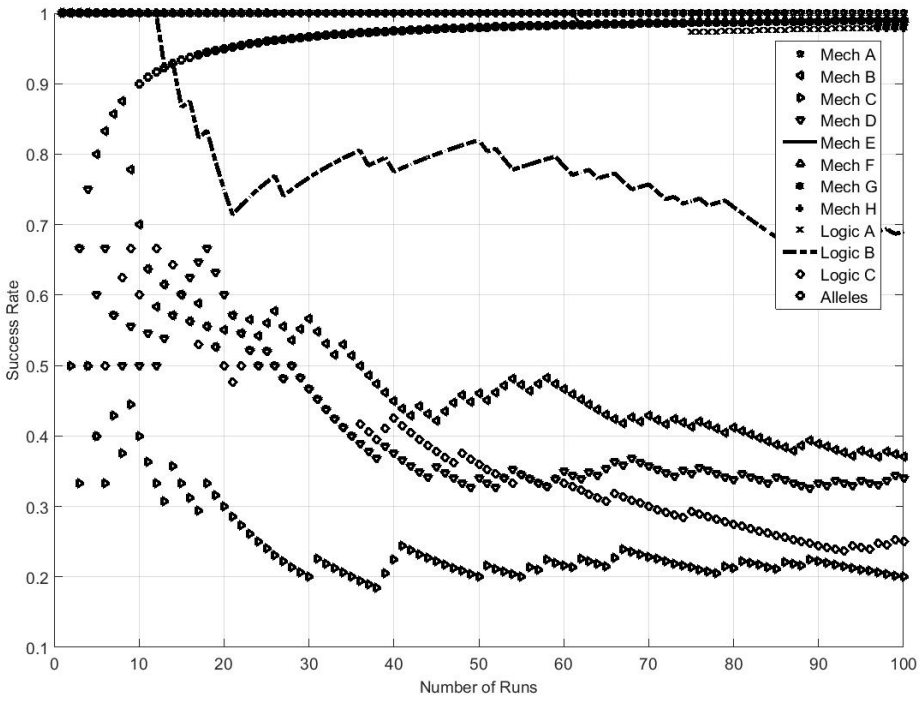


Figure 4.14: Success rate of the proposed mechanisms in the Earth to Mars problem (MGADSM model).

For the Earth to Jupiter problem, Olympio and Marmorat used the primer vector theory [75] and assumed a fixed flyby sequence as EVEJ. By setting the departure, arrival, and flyby dates fixed, a mission with four DSMs (two DSM in the first two legs and two DSM in the last leg) was obtained. The total cost of this mission was 10.267 km/s. The initial HGGA concept with feasibility criteria (original HGGA) was also tested on this problem and found a solution of cost 10.178 km/s [3]. This algorithm could find the optimal flyby sequence of EVEJ and one DSM in the second leg. The Dynamic-size multiple population GA has also been tested on this problem and the cost of its solution is 10.125 km/s [73] for EVEJ flyby sequence and one DSM in each leg of the mission. By changing the mission launch and arrival windows to a range

between 2018 and 2028, the structured-chromosome evolutionary algorithms solved the VSDS version of this problem and found a cost of 8.9134 km/s for the MGADSM phase with three flybys around Venus, Earth, and Earth (EVEEJ) and one DSM in the last leg [24]. If the duration of the mission increases, the total cost would decrease. This has been shown in the works done by Musegaas [76] and Myatt et. al. [58]. Musegaas solved the Earth to Jupiter problem as a tuning step for a mission to Saturn (EVEJS). A fixed flyby sequence and large mission duration (almost 20 years and eight months) are assumed in solving the problem. The spacecraft can have powered flybys and is captured at Jupiter. No DSMs are assumed during the trajectory and by optimizing only the event times, the cost found is 7.0144 km/s. Myatt et al. solved the same problem assuming non-powered flybys and found a solution with a cost of 7.5483 km/s [58]. The total time of the mission in this dissertation is not allowed to exceed five years and hence higher cost values are found. The minimum cost found is 10.1226 km/s using Mechanism H which is slightly lower than the solutions found by previous researches with the same mission times ranges. This solution has the flyby sequence of EVEJ with one DSM in the first leg. Other references have also solved this problem with different launch windows. For example, for a launch window between 2020 and 2030, Reference [77] has reported a minimum cost of 9.558 km/s for Earth-Earth-Jupiter flyby sequence and a minimum cost of 7.524 km/s for Earth-Earth-Earth-Jupiter flyby sequence. However, the mission topology (flyby sequence) is not optimized in their work. All the proposed mechanisms in the previous section

found the optimal flyby sequence of EVEJ. Mechanism E has the lowest cost of 10.1180 with a success rate of 94% in the MGADSM phase. Mechanism H has the second best performance. Note that the lowest cost of zero-DSM phase is for mechanism H, while mechanism E has a higher cost and lower success rate in zero-DSM phase compared to mechanism H. In the MGADSM phase, five mechanisms have a success rate of higher than 90% while resulting in a cost value of around the best solution found (0.95% maximum difference). **Figure** 4.15 shows the success rates of all the mechanisms.

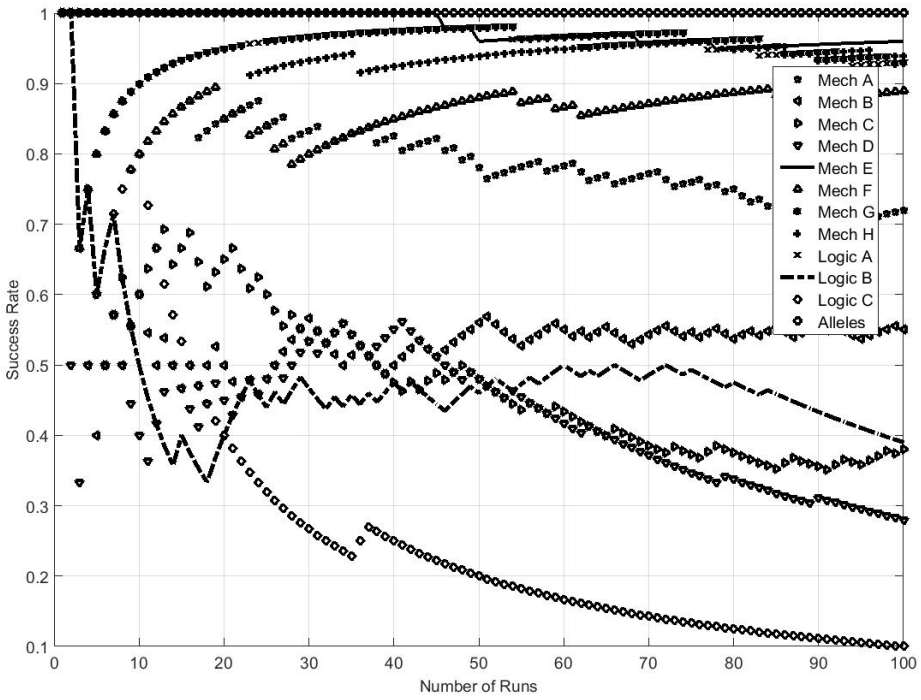


Figure 4.15: Success rate of the proposed mechanisms in the Earth to Jupiter problem (MGADSM model).

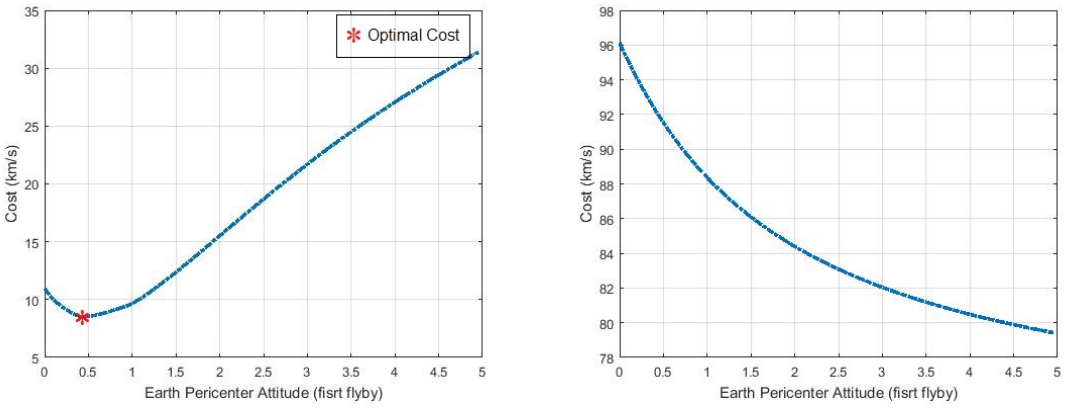
For the mission to Saturn, initial investigations show that the cost function is sensitive to the events dates (dates of performing DSMs and flybys). As an example, consider the variation of the cost function with the first flyby pericenter altitude h_p. **Figure** 4.16a shows the variation of the cost function with the pericenter altitude when all other variables are fixed at their optimal values; clearly the optimal solution corresponds to the red star in this figure. **Figure** 4.16b on the other hand shows the variation of the cost function with the pericenter altitude when the departure date is varied to a value different from its optimal value, while still keeping all other variables at their optimal values. Two observations can be noted from **Figure** 4.16. First, the impact of changing the departure date is significant on the cost; this can be depicted by comparing the cost values between the two figures (the vertical axis) with 50 days difference in their departure dates. Second, when the departure date is not optimal (**Figure** 4.16b) the line relating the cost to h_p is misleading to the optimizer. When the departure date is optimal, the cost decreases with decreasing h_p, while that is not the case when the departure date is not optimized. Hence, when optimizing the MGADSM phase, a small range is assumed around the zero-DSM variables.

The mission to Saturn has been investigated in many papers in different formats. EVEJS, Casini 1, Cassini 2 (easy and complete versions) are some of the variations on the mission that have been investigated. For the Cassini 2 (easy version), many algorithms have been tested on missions with fixed flyby sequence and DSM structures. Evolutionary algorithms and pruning techniques are applied in a mission with wide

ranges of departure and arrival dates (between 1996 to 2029) and fixed flyby sequence and DSM structure, resulting in a cost of 4.944 km/s [4]. The minimum cost reported in the literature for the VSDS version of the problem (unknown flyby sequence and DSM structure) is 8.385 km/s [3].

Schlueter developed a nonlinear mixed integer based optimization algorithm based on Ant Colony Optimization and the Oracle Penalty method. By assuming a known fixed flyby sequence, this algorithm could find a cost of 8.282 km/s [78, 79]. This problem is also solved using the parallel asynchronous generalized island model optimization (PaGMO) software using differential evolution and genetic algorithms [76]. PaGMO is an optimization software in which the user can define the problem and the optimization algorithm. The lowest cost found in this reference is 8.2379 km/s given

[4]Advances in Global Optimisation for Interplanetary Trajectory Design, https://www.esa.int/gsp/ACT/mad/projects/advancesingo.html



(a) All variables optimal **(b)** All variables optimal except t_d

Figure 4.16: Cost value vs. pericenter altitude of first flyby for Cassini 2 mission

a known fixed flyby sequence and one DSM. Other references have reported close cost values for this problem with a known fixed flyby sequence [80, 81, 82]. A list of these solutions can be found in the ESA/ACT website.

In this study, seven mechanisms were able to find the optimal flyby sequence. Mechanism H found a cost of 9.8352 km/s for this mission, with one DSM in the first leg. Despite that this cost is higher than the best known solution, the main advantage of the proposed method is its capability of searching for the optimal flyby and DSM architecture. Moreover, if by pruning the departure and arrival dates, lower cost values can be found. This has been shown in Reference [5] where by pruning the dates to a 5-day range, mechanism A results in a cost of 8.4457 km/s.

Comparing the results of different mechanism in each problem, we can see that mechanisms B, C, D, logic B, and logic C do not perform as well as other mechanisms. In these mechanisms, the number of active or hidden genes are a main factor in determining the status of the genes in the next generation. For example, in the mechanism C the children tags tend to converge to the tags of the fitter parent with higher number of hidden genes. In case of logic B and logic C, the children genes converge toward the solutions with more hidden (logic B) or more active genes (logic C). Therefore, the number of the hidden genes is determining the performance of the algorithms. However, different problems may have different number of hidden genes. An algorithm that tends to converge to solutions with higher number of hidden genes (for example

mechanism C and logic B), may not perform well when the number of hidden genes in the optimal solution is low or zero. On the other hand, some algorithms that favor higher number of hidden genes (for example mechanism B, logic C) do not perform well when the optimal solution has high number of hidden genes. In some sense, the tags' evolution in these algorithms ignores (to some level) the specifications of the problem being solved. The performance of each algorithm depends on the problem being solve and in general, we can not claim that an algorithm performs well in all the problems. However, in mechanisms A, E, F, G, H, logic A, and alleles concept, stochastic processes (crossover and mutation) have more effects on the evolution of the tags rather than the number of genes and they show better relative performance in all the tested problems. In logic A, the number of hidden tags are distributed in both children by assuming both Hidden-OR and Active-OR concepts.

Table 4.20 shows a performance comparisons of all the mechanisms in each problem. The check mark (✓) shows the mechanism that found the best objective value for each problem. $P1$ to $P8$ represent problems Egg Holder, Schwefel 2.26, Styblinski-Tang, Earth to Mars zero-DSM, Earth to Mars MGADSM, Earth to Jupiter zero-DSM, Earth to Jupiter MGADSM, and Cassini 2 MGADSM, respectively. Mechanisms H has the lowest objective value in four problems and alleles concept has the lowest objective value in three problems. However, the time consumption and success rate should be also considered when choosing suitable algorithm for a problem.

Table 4.20
Comparison of Mechanisms in test cases.

Mechanism	P1	P2	P3	P4	P5	P6	P7	P8
Mech.A								
Mech.B								
Mech.C								
Mech.D								
Mech.E							✓	
Mech.F								
Mech.G	✓		✓					
Mech.H			✓	✓		✓		✓
Logic A								
Logic B								
Logic C			✓					
Alleles		✓	✓		✓			

4.5 Conclusion

The performance of the mechanisms were tested in this chapter. The results show that the mechanisms are capable of autonomously optimizing the topology of the design space. As seen in the examples of space trajectory optimization, all the mechanisms found the (sub)optimal flyby sequence and DSMs' structures. The implementation of the new hidden genes assignment mechanisms to the space trajectory optimization problem and the mathematical optimization problems demonstrated their capability in searching for the optimal architecture, in addition to improving the solution compared to the original hidden genes genetic algorithm approach that does not implement the tags concept. It was demonstrated in this chapter that, for the trajectory optimization problems, it is possible to autonomously compute the optimal number of flybys, the planets to flyby, and the optimal number of deep space maneuvers, in

addition to the rest of the design variables using the proposed algorithms.

Statistical analysis conducted in this chapter showed that, in terms of optimality of the solution, Mechanism A, F, G, and H performed better than the other algorithms in the investigated problems. However, there is no guarantee that a mechanism works well for all the problems. Based on the problem specifications and the number of optimal hidden genes (variables), the performance of the mechanisms may vary. Even in a specific space trajectory optimization problem, the performance of the mechanisms may vary in the zero-DSM and MGADSM phases of the problem (as seen in the Earth to Jupiter problem). Therefore, an initial investigation is suggested for utilizing a specific mechanism in any application.

Chapter 5

Convergence Analysis

5.1 Introduction

In the evolutionary processes, chromosomes (solutions) evolve over subsequent iterations generating new solutions [1]. In HGGAs, the genes evolve in the same way as that in the GAs, using selection, crossover, and mutation operations. The tags have different mechanisms for evolution that were introduced in **Section** 3.2. In one mechanism, the tags evolve through stochastic operations, while in another one the tags evolve through logical operations. The performance of these mechanisms was tested on different VSDS problems, including space trajectory optimization problems in **Sections** 4.2 and 4.3. Although the HGGAs showed promising performance in the

[1]The material of this chapter are copied in part from Reference [8]

covered problems, there is no analytical proof that HGGAs with the tags evolution mechanisms are convergent. In [2], a simple implementation of a HGGA is presented where no tags are used for hiding the genes. Rather, a simple criterion is used to determine which genes are hidden in a chromosome depending on the feasibility of the solutions. Then, Holland's schema theorem [83] is implemented to prove the convergence of that simple HGGA. Some previous works on GA, however, argue that the detailed behavior of the GA can not be explained by the Schema Theorem [84, 85]. Hence, with the introduction of the new evolution mechanisms, a more comprehensive investigation of the HGGAs properties and convergence characteristics is needed.

This chapter presents a convergence analysis that proves HGGAs generate a sequence of solutions with the limit value of the global optima. For an analytical proof, the homogeneous finite Markov models of different mechanisms proposed in **Section** 3.2 are derived, and the convergence of the HGGAs with tag evolution mechanisms are investigated. The optimization problem is considered a maximization problem with strictly positive values for the objective function. In a multi-gravity-assist space trajectory optimization problem, the objective function can be defined as $1/\Delta V_{tot} > 0$, where ΔV_{tot} is proportionate to the fuel consumption. Hence, the problem can be treated as a maximization problem.

5.2 Markov Chain Model of Genetic Algorithms

The stochastic dependency between successive populations is created by applying selection, mutation, and crossover operators to the current population to produce the next population. Hence, the GA is a stochastic process in which the state of each population only depend on the state of the immediate predecessor population. Therefore, the GA can be modeled as a Markov process [86]. Several studies have investigated the convergence behavior of the GA explicitly using the Markov chain analysis [86, 87, 88, 89, 90]. The minimum conditions for convergence of ergodic GAs in the realm of Markov chain model can be found in details in [86, 90, 91]. Here, these conditions are briefly reviewed and utilized to derive the convergence conditions for the HGGAs. The GA is a Markov process and its transition matrix can be calculated. It will be shown that the GA transition matrix is reducible. Hence, the ergodic theorem for reducible transition matrix can be used to prove that ergodicity is a sufficient condition for convergence. It is assumed that this analysis is in the domain of binary genetic algorithms with bits as variables. Moreover the canonical genetic algorithms (CGAs) are considered. CGAs were proposed by Holland in 1965 and refer to GAs in which the operations of selection, crossover, and mutations are used to produce next generations. The materials of this section are a nearly verbatim adaptation of works done by Rudolph [90] and Davis [86]. We start with a review for few basic definitions:

† Column-allowable matrix: a square matrix that has at least one positive entry in each column.

† Stochastic matrix: a non-negative matrix $\mathbf{A} = (a_{ij})_{i,j=1,...,n}$ is said to be stochastic if $\sum_{j=1,...,n} a_{ij} = 1$, for each $i = 1, ..., n$.

† Arithmetic crossover: a crossover that linearly combines two parents to get one child. The child is the weighted average of the parents as follows:

$$C = \lambda P_{t_1} + (1 - \lambda) P_{t_2} \tag{5.1}$$

where C is the child, P_{t_1} and P_{t_2} are the parents, and λ is a random number in $(0, 0.5)$.

† Reducible matrix: if matrix $\mathbf{A} = (a_{ij})_{i,j=1,...,n}$ is non-negative and can be brought into the form $\begin{bmatrix} \mathbf{D} & 0 \\ \mathbf{R} & \mathbf{T} \end{bmatrix}$ by applying the same permutations to rows and columns, it is called a reducible matrix. Note that $\mathbf{D}$ and $\mathbf{T}$ should be square matrices.

The finite state space S of a Markov chain has the cardinality of $|S| = n$, where the states are numbered from 1 to n (n is the population size). Let l be the chromosome length and $M = 2^l$. $m = 2^{nl}$ is the cardinality of state space. Assume that the simple GA consists of three standard operations: selection (**S**), mutation (**M**), and crossover (**C**). To transform any state i to state j, the transition product matrix **CMS** is used

and the convergence of the GA depends on this transition matrix [91]. The transition matrix of a finite Markov chain consists of the transition probabilities from state i to j, i.e. $\mathbf{P} = (p_{ij})$. For each entry, $\sum_{j=1}^{|S|}(p_{ij}) = 1$ for all $i \in S$. The GA transition product matrix (**CMS**) is a Markov probability matrix (**P**).

First few needed theorems and lemmata are listed here:

Lemma 1: Let **C**, **M** and **S** be stochastic matrices, where **M** is positive and **S** is column-allowable. Then the product **CMS** is positive [90].

Theorem 1: Let **P** be a primitive stochastic matrix. Then $\mathbf{P}^k$ converges as $k \to \infty$ to a positive stable stochastic matrix $\mathbf{P}^\infty = 1'\mathbf{p}^\infty$, where $\mathbf{p}^\infty = \mathbf{p}^0 . \lim_{k\to\infty} \mathbf{P}^k = \mathbf{p}^0\mathbf{P}^\infty$ has nonzero entries and is unique regardless of the initial distribution [90].

Theorem 2: Let **P** be a reducible stochastic matrix defined as: $\begin{bmatrix} \mathbf{D} & 0 \\ \mathbf{R} & \mathbf{T} \end{bmatrix}$ where **D** is an $m \times m$ primitive stochastic matrix and $\mathbf{R}, \mathbf{T} \neq 0$. Then

$$\mathbf{P}^\infty = \lim_{k\to\infty} \mathbf{P}^k = \lim_{k\to\infty} \begin{bmatrix} \mathbf{D}^k & 0 \\ \sum_{i=0}^{k-1} \mathbf{T}^i\mathbf{R}\mathbf{D}^{k-i} & T^k \end{bmatrix} = \begin{bmatrix} \mathbf{D}^\infty & 0 \\ \mathbf{R}_\infty & 0 \end{bmatrix} \tag{5.2}$$

is a stable stochastic matrix with $\mathbf{P}^\infty = 1'\mathbf{p}^\infty$, where $\mathbf{p}^\infty = \mathbf{p}_0\mathbf{P}^\infty$ is unique regardless of the initial distribution, and $\mathbf{p}^\infty$ satisfies: $p_i^\infty > 0$ for $1 \leq i \leq m$ and $p_i^\infty = 0$ for $m < i \leq n$ [90].

Theorem 3: The transition matrix of the GA with mutation probability $p_m \in (0, 1)$, crossover probability $p_c \in [0, 1]$ and proportional selection is primitive [90].

Corollary 1: The CGA with parameter ranges as in Theorem 1 is an ergodic Markov chain, i.e., there exists a unique limit distribution for the states of the chain with nonzero probability to be in any state at any time regardless of the initial distribution. This is an immediate consequence of Theorems 1 and 2 [90].

Theorem 4: The CGA with parameter ranges as in Theorem 3 does not converge to the global optimum [90].

Theorem 5: In an ergodic Markov chain the expected transition time between initial state i and any other state j is finite, regardless of the states i and j [90].

Theorem 6: The canonical GA as in Theorem 3 maintaining the best solution found over time after selection converges to the global optimum [90].

To maintain the best solution over time, the population is enlarged by adding the super individual to it. The term super individual is used for the solution that does not take part in the evolutionary process. Hence, the cardinality of the state space grows from 2^{nl} to $2^{(n+1)l}$. The super individual is placed at the leftmost position in the $(n + 1)$-tuple and can be accessible by $\pi_0(i)$ from a population at state i, where $\pi_0(i)$ is a function that calls the super individual from population i.

The super individual does not take part in the evolutionary process, therefore, the extended transition matrices for crossover $\mathbf{C}^+$, mutation $\mathbf{M}^+$, and selection $\mathbf{S}^+$ can be written as [90]:

$$\mathbf{C}^+ = \begin{bmatrix} \mathbf{C} & & & \\ & \mathbf{C} & & \\ & & \ldots & \\ & & & \mathbf{C} \end{bmatrix}, \mathbf{M}^+ = \begin{bmatrix} \mathbf{M} & & & \\ & \mathbf{M} & & \\ & & \ldots & \\ & & & \mathbf{M} \end{bmatrix}, \mathbf{S}^+ = \begin{bmatrix} \mathbf{S} & & & \\ & \mathbf{S} & & \\ & & \ldots & \\ & & & \mathbf{S} \end{bmatrix} \tag{5.3}$$

Then we can write:

$$\mathbf{C}^+\mathbf{M}^+\mathbf{S}^+ = \begin{bmatrix} \mathbf{CMS} & & & \\ & \mathbf{CMS} & & \\ & & \ldots & \\ & & & \mathbf{CMS} \end{bmatrix} \tag{5.4}$$

where $\mathbf{C}^+$, $\mathbf{M}^+$, and $\mathbf{S}^+$ are block diagonal matrices and each of the 2^l square matrices $\mathbf{C}$, $\mathbf{M}$ and $\mathbf{S}$ are of size $2^{nl} \times 2^{nl}$, and $\mathbf{CMS} > 0$.

The upgrade matrix $\mathbf{U}$ is a matrix that upgrades the solutions in the population

based on their objective function value (fitness). An intermediate state containing a solution with an objective value better than the super individual will upgrade to a state where the super individual equals the better solution. Let b be the best individual of the population at state i, excluding the super individual. By definition, $u_{ij} = 1$ if $f(\pi_0(i)) < f(b)$, otherwise $u_{ii} = 1$. Therefore, there is one entry in each row and for every state j with $f(\pi_0(j)) < max[f(\pi_k(j))|k = 1 \ldots n]$, the elements will be $u_{ij} = 0$ for all is. Hence, the structure of the upgrade matrix can be written as [90]:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_{11} & & & \\ \mathbf{U}_{21} & \mathbf{U}_{22} & & \\ \ldots & \ldots & \ldots & \\ \mathbf{U}_{2^l,1} & \mathbf{U}_{2^l,2} & \ldots & \mathbf{U}_{2^l,2^l} \end{bmatrix} \tag{5.5}$$

where the sub-matrices $\mathbf{U}_{ab}$ are of size $2^{nl} \times 2^{nl}$. If the optimization problem has only one global solution, then only $\mathbf{U}_{11}$ is a unit matrix, and all other matrices $\mathbf{U}_{aa}$ with $a \geq 2$ are diagonal matrices with some zero diagonal elements, and some unit diagonal elements. Recall that in this Markov model for GA, $\mathbf{P} = \mathbf{CMS}$ and hence the transition matrix for the GA becomes:

$$\mathbf{P}^{+}=\begin{bmatrix}\mathbf{P} & & & \\ & \mathbf{P} & & \\ & & \ldots & \\ & & & \mathbf{P}\end{bmatrix}\begin{bmatrix}\mathbf{U}_{11} & & & \\ \mathbf{U}_{21} & \mathbf{U}_{22} & & \\ \ldots & \ldots & \ldots & \\ \mathbf{U}_{2^l,1} & \mathbf{U}_{2^l,2} & \ldots & \mathbf{U}_{2^l,2^l}\end{bmatrix}=\begin{bmatrix}\mathbf{PU}_{11} & & & \\ \mathbf{PU}_{21} & \mathbf{PU}_{22} & & \\ \ldots & \ldots & \ldots & \\ \mathbf{PU}_{2^l,1} & \mathbf{PU}_{2^l,2} & \ldots & \mathbf{PU}_{2^l,2^l}\end{bmatrix} \tag{5.6}$$

Note that $\mathbf{PU}_{11}=\mathbf{P}>0$. The sub-matrices $\mathbf{PU}_{a1}$, where $a\geq 2$, are gathered in a rectangular matrix $\mathbf{R}\neq 0$. Note that The $\mathbf{PU}_{1j}=0$ where $\forall j>1$. Then comparing **Equation** (5.6) to **Equation** (5.2), we can see that $\lim_{k\to\infty}\mathbf{P}^{+k}$ is unique regardless of the initial distribution, concluding in the convergence of the canonical GA.

Note that to make the extended transition matrix in the form of **Equation** (5.6), we assumed that $\mathbf{C}$, $\mathbf{M}$, and $\mathbf{S}$ are stochastic, positive, and column-allowable. Therefore, the extended transition matrices $\mathbf{C}^{+}$, $\mathbf{M}^{+}$, and $\mathbf{S}^{+}$ are stochastic and positive. The above proof also shows that the $\mathbf{P}^{+}$ in **Equation** (5.6) is a reducible matrix. Since $\mathbf{PU}_{11}>0$ ($\mathbf{PU}_{11}$ corresponding to the $\mathbf{D}$ matrix in Theorem 2), then using Theorem 2 we can show that the GA converges to the optimal solution in the limit. In section 5.3, these matrices are explicitly derived and it is shown that in the HGGA, the $\mathbf{C}$, $\mathbf{M}$, and $\mathbf{S}$ are stochastic, positive, and column-allowable.

5.3 Markov Chain Model of Hidden Genes Genetic Algorithm

The HGGAs using any of the stochastic or logical mechanisms, defined in **Chapter** 3, are here proved to be convergent. The approach to prove that these HGGA mechanisms are convergent, in general, is as follows:

> First we show that the HGGA can be modeled as a Markov process. Then it is shown that the selection, mutation, and crossover matrices have the properties described in Lemma 1. Therefore, the extended transition matrix of HGGA is reducible and can be written in the form of **Equation** (5.6). Finally, Theorem 2 can be used to prove the convergence.

Similar to the canonical GA, any future state of the HGGA population is only dependent on the current population and is independent from the previous history. Hence, if the transition product matrix **CMS** of a HGGA mechanism is stochastic, then the HGGA with that mechanism can be considered as a Markov processes.

To prove that the **CMS** matrix is stochastic and primitive, the intermediate matrices of **C**, **M** and **S** need to be derived. They are derived in this section. It is assumed that the single-point crossover is selected for the genes, unless otherwise stated. The

number of genes is L and the number of the tags is L_t. $H(i,j)$ is the Hamming distance between the genes of i and j (number of bits that must be altered by mutation to transform the *genes* of j into the *genes* of i) and is $0 \leq H(i,j) \leq L$. $H_t(i,j)$ is the Hamming distance between the tags of i and j (number of bits that must be altered by mutation to transform the *tags* of j into the *tags* of i) and is $0 \leq H_t(i,j) \leq L_t$. In all the mechanisms, the genes go thorough selection, mutation, and crossover similar to the standard genetic algorithm and only the tags evolution is different.

The transition probability matrices determine the probability of transferring a solution i to solution j; that is to change the L genes of solution i to be the same as the L genes of solution j, and change the L_t tags of solution i to be the same as the L_t tags of solution j.

5.3.1 Selection Matrix S

The selection operator for the HGGA is not different from that of a canonical GA one. For example, for a fitness proportionate selection, the probability that a solution i is selected only depends on the objective value, which in turn is a function of the values of the genes as well as the values of the tags. Hence, the selection matrix is computed for the HGGA in a similar way to that of the GA as follows.

The probability of selecting a solution $i \in S$, from a population described by the

probability distribution vector $\bar{n} \in S'$ is [86]:

$$P_1(i|\bar{n}) = \frac{n(i).R(i)}{\sum_{j \in S} n(j).R(j)} \tag{5.7}$$

where $\bar{n} = (n(0), n(1), ..., n(2^l - 1))$ is the current generation and $n(i)$ represents the number of occurrences of solution i, and $R(i)$ is the objective value for solution i and is strictly positive. Therefore, given the present generation is $\bar{n}$, the conditional probability of the successor generation $\bar{m}$ is a multinomial distribution [86]:

$$P_1(\bar{m}|\bar{n}) = \binom{M}{\bar{m}} \prod_{i \in S} P_1(i|\bar{n})^{m(i)} \tag{5.8}$$

where,

$$\binom{M}{\bar{m}} = \frac{M!}{\prod_{i \in S} (m(i)!)} \tag{5.9}$$

The transition probability matrix of the Markov chain where only the selection operation is applied is $\bar{P} = [P_1(\bar{m}|\bar{n})]$. This matrix is positive, stochastic, and column-allowable. Hence, the transition matrix due to only selection operation in HGGA is stochastic, positive, and column-allowable.

5.3.2 Mutation M and Crossover C Matrices

In this section, the explicit formulation of mutation and crossover matrices are derived and it is shown that for all of the mechanisms, the mutation matrix is stochastic and positive and the crossover matrix is stochastic. The general scheme for deriving these matrices is first presented; then followed by its application to each mechanism. Assume that the mutation probability has a nonzero value, i.e., $0 < p_m(k) \leq 1/2$. In the mutation operation in the CGA, the probability of transforming j into i can be calculated as $p_m^{H(i,j)}(1-p_m)^{L-H(i,j)}$. Thus the transition probability, due to both selection and mutation operations, is [86]:

$$P_2(i|\bar{n}) = \sum_{j \in S} p_m^{H(i,j)}(1-p_m)^{L-H(i,j)} P_1(j|\bar{n}) = \frac{1}{(1+\alpha)^L} \sum_{j \in S} \alpha^{H(i,j)} P_1(j|\bar{n}), \bar{n} \in S', i \in S \tag{5.10}$$

where $\alpha = \frac{p_m}{1-p_m}$.

$$\therefore P_2(i|\bar{n}) = \frac{\sum_{j \in S} \alpha^{H(i,j)}(n(j).R(j))}{(1+\alpha)^L . \sum_{k \in S} n(k).R(k)} \tag{5.11}$$

The multinomial distribution for $P_2(\bar{m}|\bar{n})$ can be defined as [86]:

$$P_2(\bar{m}|\bar{n}) = \binom{M}{\bar{m}} \prod_{i \in S} P_2(i|\bar{n})^{m(i)} \tag{5.12}$$

Then the transition matrix of selection and mutation would be $\bar{P} = [P_2(\bar{m}|\bar{n})]$. Note that α is positive for $0 < p_m \leq 1/2$. As can be seen from **Equation** (5.11), since α

is positive, R is positive, and $n \geq 0$, then the $\bar{P}$ matrix is primitive.

Regarding the crossover operation, assume that a single-point crossover is applied. The new function $I(i, j, k, s)$ is defined where i,j, $k \in S$, and $s \in [1, ..., L-1]$ is a bit string. The selected parents are i, j and k is a potential descendant string after a crossover at random location s which is assumed uniformly distributed. If k is produced by crossing i and j at the location s, then $I(i, j, k, s) = 1$, otherwise $I(i, j, k, s) = 0$. The conditional probability of producing k via selection and crossover operations can be derived as [86]:

$$P_2'(k|\bar{n}) = \sum_{i \in S} \sum_{j \in S} \left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L-1} \sum_{s} I(i, j, k, s) \right) + (1 - p_c).P_1(k|\bar{n}) \quad (5.13)$$

Therefore the conditional probability of producing k via selection, mutation, and crossover operations is [86]:

$$P_3(i|\bar{n}) = \frac{1}{(1+\alpha)^L} \sum_{j \in S} \alpha^{H(i,j)} P_2'(j|\bar{n}) \quad (5.14)$$

Then:

$$P_3(\bar{m}|\bar{n}) = \binom{M}{\bar{m}}. \prod_{i \in S} P_3(i|\bar{n})^{m(i)} \quad (5.15)$$

By inspection of **Equation** (5.13) and **Equation** (5.14), it can be seen that this three-operator Markov chain is primitive. Then, based on the results of section 5.2 this GA model, maintaining the best solution found over time, converges to the global

optimum.

Here, the above results are applied to each of the HGGA mechanisms.

† Mechanism A: In this mechanism, the tags can crossover independently from the genes and there is is a 10% mutation probability in the tags. This implies that the intermediate transition matrix for mutation ($\mathbf{M}$) consists of two parts, where the Hamming distance of $H(i,j)$ is the number of bits in the genes only that need to be altered by mutation, and $H_t(i,j)$ is the number of bits in the tags only that need to be altered by mutation. Hence the probability can be described as follows:

$$P_2(i|\bar{n}) = \sum_{j \in S} p_m^{H(i,j)} (1-p_m)^{L-H(i,j)} p_{mt}^{H_t(i,j)} (1-p_{mt})^{L_t - H_t(i,j)} P_1(j|\bar{n}) \qquad (5.16)$$

Note that the probability that solution j is transfered to solution i is $p_m^{H(i,j)}(1-p_m)^{L-H(i,j)}(0.1)^{H_t(i,j)}(0.9)^{L_t-H_t(i,j)} > 0$ for all $i,j \in S$ when $0 < P_m < 0.5$. Thus, $\mathbf{M}$ is positive. For the crossover operation:

$$P_2'(k|\bar{n}) = \sum_{i \in S} \sum_{j \in S} \left(P_1(i|\bar{n}) P_1(j|\bar{n}) \frac{p_c}{L-1} \frac{1}{L_t - 1} \sum_s I'(i,j,k,s,s_t) \right) + (1-p_c) P_1(k|\bar{n}) \qquad (5.17)$$

The $I'(i,j,k,s,s_t)$ takes values $\{0,1\}$, where 1 shows that child k (genes and

tags) is produced by the crossover of parents i and j at site s in the genes and at site s_t in the tags. Therefore, the conditional probability of constructing a bit string k via selection, mutation, and crossover operations in HGGA is:

$$P_3(i|\bar{n}) = \frac{1}{(1+\alpha)^{L+L_t}} \sum_{j \in S} \alpha^{H(i,j)} P_2'(j|\bar{n}) \tag{5.18}$$

Then the transition matrix for Mechanism A can be computed by substituting **Equation** (5.18) into **Equation** (5.15). Note that L is replaced by $L + L_t$ to account for the additional tags. By inspection of **Equation** (5.18), it can be concluded that this transition matrix of HGGA with mechanism A is stochastic and positive.

† Mechanism B: In this mechanism, the tags are considered as design variables in the crossover operation. The arithmetic crossover is used in this mechanism, where the number of variables in this case is $L + L_t$. Hence, it can be concluded that the crossover transition matrix $P_2'(k|\bar{n})$ (defined in **Equation** (5.13)) for mechanism B is stochastic. The mutation operation in mechanism B is similar to that of mechanism A, and hence the mutation transition matrix $P_2(i|\bar{n})$ can be computed using **Equation** (5.25) for mechanism B, which is positive when $0 < P_m < 0.5$. Finally, the $P_2(i|\bar{n})$ and $P_2'(k|\bar{n})$ matrices are used to compute $P_3(\bar{m}|\bar{n})$ using **Equations** (5.14) and (5.15). Then the overall transition matrix $P_3(\bar{m}|\bar{n})$ is primitive for mechanism B.

† Mechanism C: here an arithmetic crossover operation is used for the genes, while the tags are copied from one of the parents as described in **Section** 3.2. The selection and crossover transition probability is defined as follows:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n})P_1(j|\bar{n})p_c F_A(i,j,k,\lambda)F_{T_1}(i,j,k,f_{m_1}(i),f_{m_1}(j)) + (1-p_c)P_1(k|\bar{n}) \tag{5.19}$$

where F_A is 1 if the arithmetic crossover of genes in parents i and j, along with the weight coefficient λ result in the genes of solution k; otherwise $F_A = 0$. Also, F_{T_1} is 1 if the tags of solution k are similar to the tags of the parent that has better f_{m_1}; otherwise $F_{T_1} = 0$. For example, if parents i and j are selected and their modified cost values are $f_{m_1}(i)$ and $f_{m_1}(j)$ (defined in **Section** 3.2, Mechanism C), then if $f_{m_1}(i)$ is better than $f_{m_1}(j)$ and the tags of k are similar to the tags of i, then $F_{T_1} = 1$; otherwise $F_{T_1} = 0$. Hence, the resulting crossover probability matrix is stochastic. The Mutation operation is similar to that of mechanisms A and B, and therefore, it is stochastic and positive.

† Mechanism D: similar to mechanism C, the crossover probability can be written

as:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n})P_1(j|\bar{n})p_c F_A(i,j,k,\lambda)F_{T_2}(i,j,k,f_{m_2}(i),f_{m_2}(j)) \\ +(1-p_c)P_1(k|\bar{n}) \tag{5.20}$$

where F_A is 1 if the arithmetic crossover of genes in parents i and j along with weight the coefficient λ result in the genes of solution k; otherwise $F_A = 0$. Also, F_{T_2} is 1 if the tags of solution k are similar to the tags of the parent that has better f_{m_2}; otherwise $F_{T_2} = 0$. Hence, the resulting crossover probability matrix is stochastic. The Mutation operation is similar to that of mechanisms A and B, and therefore, it is stochastic and positive.

† Mechanism E: tags evolve through a mutation operation with a certain mutation probability. Let p_{mt} be the mutation probability of the tags, then:

$$P_2(i|\bar{n}) = \sum_{j\in S} p_m^{H(i,j)}(1-p_m)^{L-H(i,j)} p_{mt}^{H_t(i,j)}(1-p_{mt})^{L_t-H_t(i,j)} P_1(j|\bar{n}) \tag{5.21}$$

which is stochastic. Also since p_m and p_{mt} are positive and less than 0.5, then $P_2(i|\bar{n})$ is positive. The crossover is only applied to the genes in this mechanism,

hence:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S}\left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L-1}\sum_s I(i,j,k,s)\right) + (1-p_c).P_1(k|\bar{n}) \tag{5.22}$$

Similar to the CGA, the matrix $P_2'(k|\bar{n})$ above is stochastic.

† Mechanism F: In this mechanism, the tags are considered as discrete variables similar to the design variables in the chromosome. The crossover and mutation operations are performed on all the variables (genes and tags). The mutation transition probability is then as follows:

$$P_2(i|\bar{n}) = \sum_{j\in S} p_m^{H(i,j)+H_t(i,j)}(1-p_m)^{L+L_t-H(i,j)-H_t(i,j)}P_1(j|\bar{n}) \tag{5.23}$$

which results in a positive and stochastic mutation matrix. Also the stochastic crossover transition probability can be calculated as follows:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S}\left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L+L_t-1}\sum_s I(i,j,k,s)\right) + (1-p_c).P_1(k|\bar{n}) \tag{5.24}$$

† Mechanism G: In this mechanism, the tags are considered as discrete variables similar to the design variables in the chromosome; yet only the crossover operation is applied to the tags. Since there is no mutation in the tags, the mutation

transition probability is as follows:

$$P_2(i|\bar{n}) = \sum_{j \in S} p_m^{H(i,j)} (1 - p_m)^{L + L_t - H(i,j)} P_1(j|\bar{n}) \tag{5.25}$$

which results in a positive and stochastic mutation matrix. The stochastic crossover probability matrix is similar to **Equation** (5.24).

† Mechanism H: In this mechanism, the tags are considered as discrete variables similar to the design variables in the chromosome; yet only the mutation operation is applied to the tags. Hence, the mutation matrix is similar to **Equation** (5.23) which is stochastic and positive. The crossover probability matrix is similar to **Equation** (5.22); which is stochastic.

† Alleles: In this concept, the HGGA is developed by simulating alleles and considering two tags for each gene, one recessive and one dominant. The alleles go through mutation and crossover.

Let the length of the alleles be $2L_t$, and let H_a be the Hamming distance between the tags of the i and j alleles (number of bits that must be altered by mutation to transform the tags of j into the tags of i). The maximum of H_a is $2L_t$. Since all the bits go through mutation with probability p_m, the mutation conditional probability can be calculated as:

$$P_2(i|\bar{n}) = \sum_{j \in S} p_m^{H(i,j)+H_a(i,j)} (1 - p_m)^{L + 2L_t - H(i,j) - H_a(i,j)} P_1(j|\bar{n}) \tag{5.26}$$

which results in a stochastic and positive mutation matrix. There are two crossover points, one in the genes and one in the tags such that $s_t \in [1, ..., L_t - 1]$. The crossover points in tags (s_t) are similar in the dominant and recessive alleles. Hence:

$$P_2'(k|\bar{n}) = \sum_{i \in S} \sum_{j \in S} \left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L-1}.\frac{1}{L_t - 1} \sum_s I'(i,j,k,s,s_t) \right) + (1 - p_c).P_1(k|\bar{n}) \quad (5.27)$$

where $I'(i,j,k,s,s_t)$ is 1 if the crossover of i and j at site s in genes and site s_t in tags produce k, otherwise $I'(i,j,k,s,s_t) = 0$. The crossover matrix in **Equation** (5.27) is stochastic.

† Logic A: the member of the current generation ($\bar{n}$) is split into two groups of equal size. For the first group, the Hidden-Or logic is applied on the tags and for the other half, the Active-Or logic is used in the tags. There is no mutation in the tags; hence the mutation probability matrix is defined as in **Equation** (5.25). Let F_{HO} and F_{AO} be functions that can have values of 0 or 1. If the Hidden-Or operator on the tags of i and j results in the tags of k, then $F_{HO}(i,j,k) = 1$, otherwise $F_{HO}(i,j,k) = 0$. If the Active-Or operator on the tags of i and j results in the tags of k, then $F_{AO}(i,j,k) = 1$, otherwise $F_{AO}(i,j,k) = 0$. For the first half of the children the crossover probability

matrix is then:

$$P_2'(k|\bar{n}_1) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}_1).P_1(j|\bar{n}_1).F_{HO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}_1) \tag{5.28}$$

and for the second half of the children:

$$P_2''(k|\bar{n}_2) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}_2).P_1(j|\bar{n}_2).F_{AO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}_2) \tag{5.29}$$

Where $\bar{n}_1$ represents one half of the GA search space, and $\bar{n}_2$ represents the other half of the GA search space. The conditional probability of producing k with i and j via selection and crossover is $P_2'(k|\bar{n}_1) \times P_2''(k|\bar{n}_2)$, which results in a stochastic matrix.

† Logic B: The Hidden-OR logic is used for both children. Even though the tags will be the same in both children, the two children represent two different solutions because they have different gene values. There is no mutation for the tags, hence, the mutation probability matrix is defined as in **Equation** (5.25). The crossover probability matrix is:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}).P_1(j|\bar{n}).F_{HO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}) \tag{5.30}$$

Both mutation and crossover matrices are stochastic; in addition the mutation conditional probability is positive.

† Logic C: The Active-OR logic is used for both children. Even though the tags will be the same in both children, the two children represent two different solutions because they have different gene values. The mutation probability matrix is defined as in **Equation** (5.25). The crossover probability matrix is:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}).P_1(j|\bar{n}).F_{AO}(i,j,k).\frac{p_c}{L-1}\sum_{s} I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}) \tag{5.31}$$

Both mutation and crossover matrices are stochastic; in addition the mutation conditional probability is positive.

By calculating the **C**, **M**, and **S** matrices of different mechanisms, we can now continue on the convergence analysis. As shown, the mutation matrices in all the mechanisms are stochastic and positive. The selection matrix is also stochastic and positive; and hence it is column-allowable. Also the crossover matrices are stochastic. Hence, the **CMS** matrix is positive (Lemma 1). Since the HGGA maintains the best solution found over time after selection, Theorem 6 can be used to prove that all mechanisms of HGGA presented above are convergent.

5.4 Numerical Analysis

The results of the previous section show that the HGGAs using any of the proposed tags evolution mechanisms are convergent. Here, the numerical convergence is investigated on interplanetary trajectory optimization problems. In **Chapter** 4, it was demonstrated that HGGAs can search for optimal solution architectures, and find the optimal topology for bench mark interplanetary trajectory optimization problems.

As example, the convergence of logic A in the zero-DSM phase of Earth to Mars problem is shown in **Figure** 5.1. The simulation are repeated 100 times and the cost value in each generation of each simulation is calculated in each generation. To have a clear figure, Only 5 simulation are shown (chosen randomly). As shown, the algorithm is convergent in this problem.

The convergence of mechanism D in the MGADSM phase of Earth to Jupiter problem is shown in **Figure** 5.2. The simulations are repeated 100 times and again, this algorithm show a convergent behavior (only 5 simulations are shown).

Other mechanisms also show convergent behavior in all the tested problems; including mathematical and space trajectory optimization problems (zero-DSM and MGADSM phases).

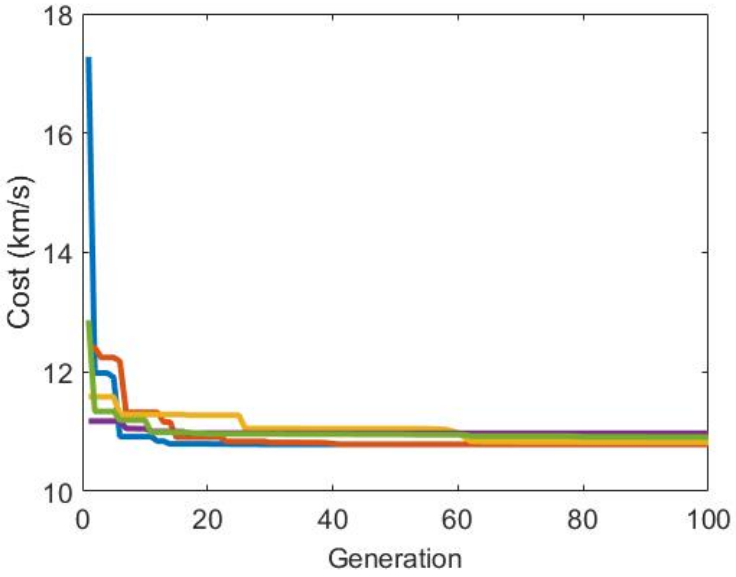


Figure 5.1: Numerical Convergence of 5 simulations in Earth to Mars problem (zero-DSM model) using logic A.

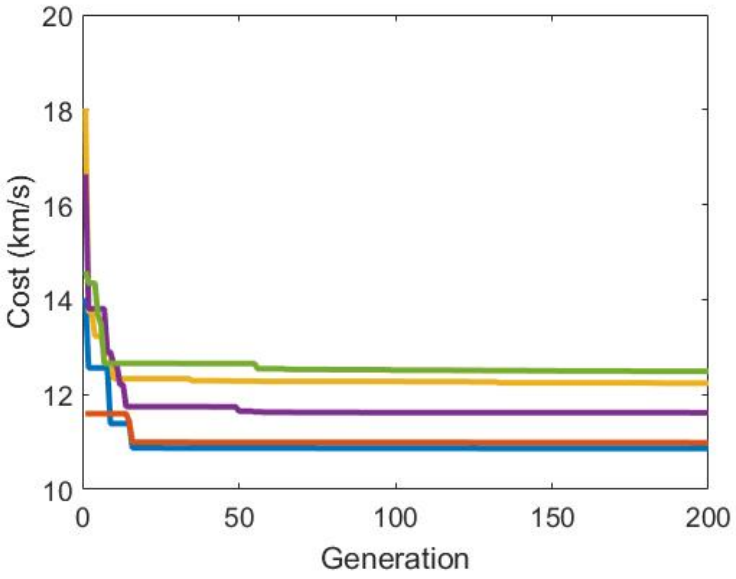


Figure 5.2: Numerical Convergence of 5 simulations in Earth to Jupiter problem (MGADSM model) using mechanism D.

Chapter 6

Conclusion

6.1 Dissertation Summary and Conclusion

Optimization of problems with variable number of design variables can be a complicated task especially when the number of design variables increase. In this dissertation novel optimization algorithms are proposed for these types of problems. The biologically inspired HGGAs are developed to enable the creation of feasible solutions with variable length of variables. HGGAs are based on GAs in which the design variables are modeled as genes. In the first chapter, the required background on optimization algorithms and VSDS problems are presented and in the second chapter, the concept of GAs and HGGAs are explained. The initial method of implementing HGGAs,

which was proposed in literature, is presented and its limitations are presented. In HGGAs, the genes can be hidden or active. Hidden genes do not affect the objective function of the optimization but take part in the evolutionary process to produce the next generation. By hiding some genes in each chromosome, different solutions with different number of variables can be obtained. In the initial method of implementing HGGAs, genes are hidden one by one from one side until a feasible solution is achieved. New concepts for HGGAs are developed in chapter three. In these concepts, binary tags are assigned to the gens that can determine if the corresponding gene should be hidden or not. Different evolutionary mechanisms are developed for these tags. These evolutionary mechanisms autonomously determine the status of the genes and can be implemented to any problem easily.

The proposed algorithms are tested on three mathematical problems and their performance is evaluated compared to initial method of implementing HGGAs and standard GAs. As an example to show the efficiency of the algorithms on more complex problems, three interplanetary trajectory optimization problems are also tested. The results are compared to other heuristic algorithms and the initial method of implementing HGGAs. The results show that the proposed HGGAs can be successfully utilized to search for optimal solutions as well as the optimal architecture of the solutions. In some cases, the algorithms perform better than other excising methods.

Moreover, the convergence of these algorithms are mathematically and numerically

analyzed. Markov chain matrices are derived for all the proposed mechanisms. Their ergodicity and elitism conditions are investigated and it is proven that all the proposed mechanisms are convergent. Numerical analysis are also performed on the investigated mathematical and interplanetary trajectory problems and their convergence are graphically presented.

The proposed algorithms use general concepts and can be used for any VSDS optimization problem including constrained or non-constrained, discontinuous, non-differentiable, stochastic, or highly nonlinear problems. Some algorithms including mechanism E, F, and H showed better performance in the investigated problems. However, the results of the tested problems can not be generalized to other problems and to select the best mechanism for any problem, an initial investigation is suggested.

For future studies, the convergence rate of the proposed mechanisms can be analytically investigated. In any problem, it is important and beneficial to know how fast the algorithm converges to the optimum. The convergence rate is defined as the normalized mean of the change in the objective function value over a generation. The lower bounds of the convergence rate has been previously calculated in Markov chain models of evolutionary algorithms. Based on the transition matrices derived in this dissertation, the lower bounds of the convergence rates can be calculated for each mechanism. It is interesting to compare the analytical rates to the results of

References

[1] The National Institute of General Medical Sciences, "The New Genetics." `http://publications.nigms.nih.gov/thenewgenetics/chapter2.html`, 2018. Online; accessed 04 April 2018.

[2] O. Abdelkhalik, "Hidden genes genetic optimization for variable-size design space problems," *Springer, Journal of Optimization Theory and Applications*, vol. 156, no. 2, pp. 450–468, 2012. doi: 10.1007/s10957-012-0122-6.

[3] A. Gad and O. Abdelkhalik, "Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization," *AIAA Journal of Spacecraft And Rockets*, vol. 48, no. 4, pp. 629–641, 2011. doi: 10.2514/1.52642.

[4] O. Abdelkhalik and S. Darani, "Evolving hidden genes in genetic algorithms for systems architecture optimization," *Journal of Dynamic Systems, Measurement and Control*, 2018. accepted for publication.

[5] S. Darani and O. Abdelkhalik, "Space trajectory optimization using hidden

genes genetic algorithms," *Journal of Spacecraft and Rockets*, 2018. doi: 10.2514/1.A33994.

[6] O. Abdelkhalik and S. Darani, "Hidden genes genetic algorithms for systems architecture optimization," in *In Proceedings of the Genetic and Evolutionary Computation Conference*, (Denver, Colorado, USA), pp. 629–636, 20–24 July 2016. doi: 10.1145/2908812.2908819.

[7] S. Darani and O. Abdelkhalik, "Developments on the optimization of interplanetary trajectories using hidden genes genetic algorithms," in *AIAA/AAS Astrodynamics Specialist Conference*, (Long Beach, California, USA), 13–16 September 2016. doi: 10.2514/6.2016-5264.

[8] S. Darani and O. Abdelkhalik, "Convergence analysis of hidden genes genetic algorithms in space trajectory optimization," *Journal of Aerospace Information Systems*, 2018. doi: 10.2514/1.I010564.

[9] "National science foundation. cyber physical systems program.." `https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286`. Accessed: 2018-04-04.

[10] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Longman Publishing Co., 1989. pp. 62–84.

[11] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 27 November–1 December 1995. doi: 10.1109/ICNN.1995.488968.

[12] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, pp. 29–41, 1996. doi: 10.1109/3477.484436.

[13] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: a Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag Berlin Heidelberg, 2005.

[14] N. Cramer, "A representation for the adaptive generation of simple sequential programs," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 183–187, 1985.

[15] R. Forsyth, "Beagle–a darwinian approach to pattern recognition," *Kybernetes*, vol. 10, no. 3, pp. 159–166, 1981. doi: 10.1108/eb005587.

[16] I. Harvey, *Toward a Practice of Autonomous Systems: Proceedings of First European Conference on Artificial Life*. MIT Press Cambridge, MA, USA, 1992.

[17] I. Harvey, *The SAGA Cross the Mechanics of Recombination for Species with Variable Length Genotypes*. Parallel Problem Solving from Nature 2, North Holland, 1992.

[18] C. Lee and E. Antonsson, "Variable length genomes for evolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00), Las Vegas, Nevada, USA*, p. 806, July 8-12 2000.

[19] R. Zebulum, M. Vellasco, and M. Pacheco, "Variable Length Representation in Evolutionary Electronics," *Evolutionary Computation*, vol. 1, pp. 93–120, 2000. doi: 10.1162/106365600568112.

[20] J. Ryoo and P. Hajela, "Handling Variable String Lengths in GA-based Structural Topology Optimization," *Structural and Multidisciplinary Optimization*, vol. 26, pp. 318–325, 2004. doi: 10.1007/s00158-003-0307-6.

[21] I. Kim and O. de Weck, "Variable Chromosome Length Genetic Algorithm for Structural Topology AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Palm Springs, California, pp. 1–12, 19–22 April 2004.

[22] I. Kim and O. de Weck, "Variable Chromosome Length Genetic Algorithm for Progressive Refinement in Topology Optimization," *Structural and Multidisciplinary Optimization*, vol. 26, pp. 445–456, 2005. doi: 10.1007/s00158-004-0498-5.

[23] H. Nyew, O. Abdelkhalik, and N. Onder, "Structured chromosome evolutionary algorithms for multi-gravity-assist trajectories optimization," in *AAS/AIAA Astrodynamics Specialist Conference*, no. AIAA 2012–4519, (Minneapolis, MN, USA), August 12 – 16 2012. doi: 10.2514/1.I010272.

[24] H. Nyew, O. Abdelkhalik, and N. Onder, "Structured-chromosome evolutionary algorithms for variable-size autonomous interplanetary trajectory planning

optimization," *Journal of Aerospace Information Systems*, pp. 1–15, 2015. doi: 10.2514/1.I010272.

[25] A. Khetan, D. Lohan, and J. Allison, "Managing variable-dimension structural optimization problems using generative algorithms," *Structural and Multidisciplinary Optimization*, vol. 52, pp. 695–715, 2015. doi: 10.1007/s00158-015-1262-8.

[26] X. Guo, W. Zhang, and W. Zhong, "Topology optimization based on moving deformable components: A new computational framework," *Journal of Applied Mechanics*, vol. 81, 2014. doi: 10.1115/1.4027609.

[27] L. Yang, *Topology Optimization of Nanophotonic Devices*. Phd, Technical University of Denmark, Department of Photonics Engineering, Building 343, DK-2800 Kongens Lyngby, Denmark, 2011.

[28] K. Suzuki and N. Kikuchi, "A homogenization method for shape and topology optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 93, pp. 291–318, 1991. doi: 10.1016/0045-7825(91)90245-2.

[29] O. Abdelkhalik and A. Gad, "Dynamic-size multiple populations genetic algorithm for multigravity-assist trajectories optimization," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 520–529, 2012. doi: 10.2514/1.54330.

[30] W. Hohmann, *Erreichbarkeit der himmelskorper*. Munich and Berlin: R. Oldenbourg, 1925.

[31] J. Cornelisse, "Trajectory analysis for interplanetary missions," *ESA Journal*, vol. 2, pp. 131–143, 1978.

[32] J. Navagh, "Optimizing interplanetary trajectories with deep space maneuvers," Tech. Rep. CR4546, NASA Langley Research Center, No. CR4546, 1993, 1993.

[33] S. Molenaar, "Optimization of interplanetary trajectories with deepspace maneuvers - model development and application to a uranus orbiter mission," Master's thesis, Delft University of Technology, the Netherlands, 2009.

[34] J. Bryan, "Global optimization of mga-dsm problems using the interplanetary gravity assist trajectory optimizer (igato)," Master's thesis, California Polytechnic State University, San Luis Obispo, 2011.

[35] P. DiLizia and G. Radice, "Advanced global optimisation tools for mission analysis and design," rept. 18139/04/nl/mv, European Space Agency, 2004.

[36] M. Vasile and P. de Pascale, "Preliminary design of multiple gravity-assist trajectories," *Journal of Spacecraft and Rockets*, vol. 43, no. 4, pp. 794–805, 2006. doi: 10.2514/1.17413.

[37] C. Chilan and B. Conway, "Automated design of multiphase space missions using

hybrid optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 5, pp. 1410–1424, 2013. doi: 10.2514/1.58766.

[38] M. Vasile, L. Summerera, and P. D. Pascale, "Design of earth - mars transfer trajectories using evolutionary-branching technique," *Acta Astronautica*, vol. 56, pp. 705 – 720, 2005. doi: 10.1016/j.actaastro.2004.12.002.

[39] B. Dachwald, "Optimisation of solar sail interplanetary trajectories using evolutionary neurocontrol," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 66–72, 2004. doi: 10.2514/1.9286.

[40] M. Sentinella and L. Casalino, "Cooperative evolutionary algorithm for space trajectory optimization," *Celestial Mechanics and Dynamical Astronomy, Springer*, vol. 105, no. 1, pp. 105–211, 2009. doi: 10.1007/s10569-009-9223-4.

[41] J. Labroquere, A. Heritier, A. Riccardi, and D. Izzo, "Evolutionary constrained optimization for a jupiter capture," *Parallel Problem Solving from Nature, PPSN XIII, Lecture Notes in Computer Science*, vol. 8672, 2014. doi: 10.1007/978-3-319-10762-2_26.

[42] D. Gondelach and R. Noomen, "Hodographic-shaping method for low-thrust interplanetary trajectory design," *AIAA Journal of Spacecraft and Rockets*, vol. 52, no. 3, pp. 728–738, 2015. doi: 10.2514/1.A32991.

[43] P. D. Pascale and M. Vasile, "Preliminary design of low-thrust multiple gravity-assist trajectories," *AIAA Journal of Spacecraft and Rockets*, vol. 43, no. 5, pp. 1065–1076, 2006. doi: 10.2514/1.19646.

[44] J. Olympio and C. Yam, "Deterministic method for space trajectory design with mission margin constraints," in *61st International Astronautical Congress*, no. IAC-10.C1.9.12, (Prague, CZ), 2010.

[45] J. Liu and G. Dai, "The application of evolution-branching algorithm on earth-mars transfer trajectory," in *Computational Intelligence and Intelligent Systems, ISICA*, vol. 51, (Springer, Berlin, Heidelberg), 2009. doi: 10.1007/978-3-642-04962-0_29.

[46] L. Simoes, D. Izzo, E. Haasdijk, and A. Eiben, "Multi-rendezvous spacecraft trajectory optimization with beam p-aco," *Evolutionary Computation in Combinatorial Optimization, EvoCOP 2017*, vol. 10197, pp. 141–156, 2017. doi: 10.1007/978-3-319-55453-2_10.

[47] G. Hughes and C. McInnes, "Solar sail hybrid trajectory optimization for non-keplerian orbit transfers," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 3, pp. 602–604, 2002. doi: 10.2514/2.4924.

[48] O. Abdelkhalik and D. Mortari, "Orbit design for ground surveillance using genetic algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1231–1235, 2006. doi: 10.2514/1.16722.

[49] G. Rauwolf and V. Coverstone-Carroll, "Near-optimal low-thrust orbit transfers generated by a genetic algorithm," *Journal of Spacecraft and Rockets*, vol. 33, no. 6, pp. 859–862, 1996. doi: 10.2514/3.26850.

[50] S. Wagner, *Automated trajectory design for impulsive and low thrust interplanetary mission analysis*. PhD thesis, Iowa State University, Ames, Iowa, 2014.

[51] D. Izzo, D. Hennes, and A. Riccardi, "Constraint handling and multi-objective methods for the evolution of interplanetary trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 792–800, 2015. doi: 10.2514/1.G000619.

[52] D. Izzo, "Advances in global optimisation for space trajectory design," in *Proceedings of the 25th International Symposium on Space Technology and Science*, (Japan Society for Aeronautical and Space and ISTS), 2006.

[53] H. Shen, J. Zhou, Q. Peng, H. Li, and J. Li, "Multi-objective interplanetary trajectory optimization combining low-thrust propulsion and gravity-assist maneuvers," *Science China: Technological Sciences*, vol. 55, no. 3, pp. 841–847, 2012. doi: 10.1007/s11431-011-4705-5.

[54] D. Izzo, L. F. Simoes, M. Martens, G. de Croon, A. Heritier, and C. Yam, "Search for a grand tour of the jupiter galilean moons," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation, ACM*, (Amsterdam, The Netherlands), 2013. doi: 10.1145/2463372.2463524.

[55] M. Ceriotti, *Global optimisation of multiple gravity assist trajectories*. PhD thesis, University of Glasgow, Glasgow, Scotland, United Kingdom, 2010.

[56] N. Strange and J. Longuski, "Graphical method for gravity-assist trajectory design," *Journal of Spacecraft and Rockets*, vol. 39, no. 1, pp. 9–16, 2002. doi: 10.2514/2.3800.

[57] D. Izzo, V. Becerra, D. Myatt, S. Nasuto, and J. Bishop, "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories," *Journal of Global Optimization*, vol. 38, pp. 283–296, 2007. doi: 10.1007/s10898-006-9106-0.

[58] D. Myatt, V. Becerra, S. Nasuto, J. Bishop, and D. Izzo, "Advanced global optimisation for mission analysis and design," Tech. Rep. 03-4101a, European Space Agency, Ariadna Final Report, 2004.

[59] J. Englander, B. Conway, and T. Williams, "Automated mission planning via evolutionary algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1878–1887, 2012. doi: 10.2514/1.54101.

[60] F. Bernelli-Zazzera, M. Lavagna, R. Armellin, P. D. Lizia, F. Toppu, and M. Berz, "Global trajectory optimisation: Can we prune the solution space when considering deep space maneuvers?," Tech. Rep. 064101, European Space Agency, Ariadna Final Report, 2007.

[61] J. Miller and C. Weeks, "Application of tisserands criterion to the design of gravity assist trajectories," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2002. doi: 10.2514/6.2002-4717.

[62] A. Labunsky, O. Papkov, and K. Sukhanov, *Multiple gravity assist interplanetary trajectories*. Newark, NJ: Gordon and Breach Science Publishers, 1998.

[63] J. Smith and T. Fogarty, "Adaptively parameterised evolutionary systems: Self adaptive recombination and mutation in a genetic algorithm," *Parallel Problem Solving from Nature – PPSN IV.*, vol. 1141, pp. 441–450, 1996. doi: 10.1007/3-540-61723-X_1008.

[64] J. Smith, M. Bartley, and T. Fogarty, "Microprocessor design verification by two-phase evolution of variable length tests," in *IEEE International Conference on Evolutionary Computation*, 13–16 April.

[65] T. Jenuwein and C. Allis, "Translating the histone code," *Journal of Science*, vol. 293, pp. 1074–1080, 2001. doi: 10.1126/science.1063127.

[66] I. Tanev and K. Yuta, "Epigenetic programming: Genetic programming incorporating epigenetic learning through modification of histones," *Information Sciences*, vol. 178, pp. 4469–4481, 2008. doi: 10.1016/j.ins.2008.07.027.

[67] B. Starr, "Spooled DNA and Hidden Genes: The Latest Finding in How our DNA is Organized and Read." `http://genetics.thetech.org/original_`

news/news31, 2018. The Tech Museum of Innovation, Department of Genetics, Stanford School of Medicine, 201 South Market Street San Jose, CA 95113, Online; accessed 04 April 2018.

[68] H. Hasan, *Mendel and the Laws of Genetics*. The Rosen Publishing Group, 2005.

[69] P. Bowler, *Evolution: the history of an idea*. Berkeley, University of California Press, 2003.

[70] M. Jamil and X. Yang, "A literature survey of benchamrk functions for global optimization problems," *International Journal of Mathematical Modeling and Numerical Optimizatin*, vol. 4, no. 2, 2013. doi: 10.1504/IJMMNO.2013.055204.

[71] J. Olympio, "Designing optimal multi-gravity-assist trajectories with free number of impulses," *21st International Symposium on Space Flights Dynamics*, 2009.

[72] I. Takuto, J. Hoffman, and O. de Weck, "Method for rapid interplanetary trajectory analysis using v maps with flyby options," *Journal of the British Interplanetary Society*, vol. 64, pp. 204–213, 2011. http://hdl.handle.net/1721.1/99890, Accessed: November 2017.

[73] O. Abdelkhalik and A. Gad, "Dynamic-size multiple populations genetic algorithm for multigravity-assist trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 520–529, 2012. doi: 10.2514/1.54330.

[74] D. Matson, L. Spilker, and J. Lebreton, "The cassini/huygens mission to the saturnian system," *Space Science Reviews*, vol. 104, pp. 1–58, 2002. doi: 10.1023/A:1023609211620.

[75] J. Olympio and J. Marmorat, "Global trajectory optimization: Can we prune the solution space when considering deep space maneuvers?," September 2007. AO-1-5180-06-4101.

[76] P. Musegaas, "Optimization of space trajectories including multiple gravity assists and deep space maneuvers," Master's thesis, Delft University of Technology, the Netherlands, 2012.

[77] C. H. Yam, D. D. Lorenzo, and D. Izzo, "Constrained global optimization of low-thrust interplanetary trajectories," in *Evolutionary Computation (CEC), IEEE Congress on*, (Barcelona, Spain), 18–23 July 2010. doi: 10.1109/CEC.2010.5586019.

[78] M. Schlueter, "Nonlinear mixed integer based optimization technique for space applications," Master's thesis, The University of Birmingham, Birmingham, United Kingdom, 2012.

[79] M. Vasile, E. Minisci, and M. Locatelli, "An inflationary differential evolution algorithm for space trajectory optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 267–281, 2011. doi: 10.1109/TEVC.2010.2087026.

[80] B. Addis, A. Cassioli, M. Locatelli, and F. Schoen, "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, vol. 48, pp. 635–652, 2011. doi: 10.1007/s10589-009-9261-6.

[81] G. Danoy, B. Dorronsoro, and P. Bouvry, "New state-of-the-art results for cassini2 global trajectory optimization problem," in *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1–6, 2011. `http://hdl.handle.net/10993/16758`, Accessed: November 2017.

[82] T. Vinko, D. Izzo, and C. Bombardelli, "Benchmarking different global optimisation techniques for preliminary space trajectory design," in *58th International Astronautical Congress*, (Hyderabad, India), 24–28 September 2007.

[83] J. Holland, *Adaption in Natural and Artificial Systems.* Ann Arbor, Michigan: University of Michigan, 1975.

[84] H. Muhlenbein, *Evolutionary Algorithms: Theory and Application*, vol. 5. Lecture Series on Computer and Computational Sciences, Wiley, 2005.

[85] H. Muhlenbein, "Evolution in time and space, the parallel genetic algorithm," *Foundation of the Genetic Algorithm*, vol. 1, pp. 316–337, 1991. doi: 10.1007/BF01530781.

[86] T. Davis and J. Principe, "A markov chain framework for the simple genetic algorithm," *Evolutionary Computation*, vol. 1, pp. 269–288, 1993. doi: 10.1162/evco.1993.1.3.269.

[87] M. Vose and G. Liepins, "Punctuated equilibria in genetic search," *Complex Systems*, vol. 5, pp. 31–44, 1991.

[88] A. Nix and M. Vose, "Modeling genetic algorithms with markov chains," *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 79–88, 1992. doi: 10.1016/B978-0-08-050684-5.50023-9.

[89] J. Suzuki, "A markov chain analysis on simple genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, pp. 655–659, 1995. doi: 10.1109/21.370197.

[90] G. Rudolph, "Convergence analysis of canonical genetic algorithm," *IEEE Transaction on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994. doi: 10.1109/72.265964.

[91] A. Agapie, "Genetic algorithms: Minimal conditions for convergence," in *In Proceedings of the 2005 European Conference on Artificial Evolution*, (Heidelberg, Berlin, Germany), pp. 181–193, 2005. doi: 10.1007/BFb0026600.

[92] U. O'Reilly and F. Oppacher, "Program Search with a Hierarchical Variable Length Representation: Genetic Programming, Simulated Annealing and Hill Climbing," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, PPSN III, pp. 397–406, 1994. doi: 10.1007/3-540-58484-6_283.

[93] P. Lizia and G. Radice, "Advanced global optimisation for mission analysis and design," Tech. Rep. 18139-04-NL-MV, European Space Agency, Ariadna Final Report, 2004.

Appendix A

Copyright Permissions

The copyright permissions and correspondences for the papers used in this thesis are provided in this section.

ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Apr 06, 2018

This Agreement between Ms. Shadi Darani ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") consists of your license details and the terms and conditions provided by Association for Computing Machinery, Inc. and Copyright Clearance Center.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	4323200976947
License date	Apr 06, 2018
Licensed Content Publisher	Association for Computing Machinery, Inc.
Licensed Content Publication	Proceedings
Licensed Content Title	Hidden Genes Genetic Algorithms for Systems Architecture Optimization
Licensed Content Author	Ossama Abdelkhalik, et al
Licensed Content Date	Jul 20, 2016
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	Yes
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Order reference number	
Title of your thesis/dissertation	System Architecture Optimization Using Hidden Genes Genetic Algorithms with Applications in Space Trajectory Optimization
Expected completion date	May 2018
Estimated size (pages)	120
Requestor Location	Ms. Shadi Darani 112 Quincy St Apt E HANCOCK, MI 49930 United States Attn: Ms. Shadi Darani
Billing Type	Credit Card
Credit card info	Visa ending in 6289
Credit card expiration	03/2019
Total	8.00 USD
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

publisher reserves the right to take any and all action to protect its copyright in the materials.
11. ACM makes no representations or warranties with respect to the licensed material and adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.
12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.
13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.
14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).
15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.
16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.
17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com
18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.
Special Terms:

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

American Inst of Aeronautics and Astronautics (AIAA) LICENSE TERMS AND CONDITIONS

Apr 09, 2018

This is a License Agreement between Ms. Shadi Darani ("You") and American Inst of Aeronautics and Astronautics (AIAA) ("American Inst of Aeronautics and Astronautics (AIAA)") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by American Inst of Aeronautics and Astronautics (AIAA), and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	4324851228990
License date	Apr 06, 2018
Licensed content publisher	American Inst of Aeronautics and Astronautics (AIAA)
Licensed content title	AIAA SPACE 2016
Licensed content date	Jan 1, 2016
Type of Use	Thesis/Dissertation
Requestor type	Author of requested content
Format	Print, Electronic
Portion	chapter/article
Number of pages in chapter/article	12
The requesting person/organization is:	Shadi A. Darani
Title or numeric reference of the portion(s)	The purpose of interplanetary trajectory ... resources to compare the results to.
Title of the article or chapter the portion is from	N/A
Editor of portion(s)	N/A
Author of portion(s)	N/A
Volume of serial or monograph.	N/A
Page range of the portion	1-12
Publication date of portion	30/05/2018
Rights for	Main product
Duration of use	Life of current and all future editions
Creation of copies for the disabled	no
With minor editing privileges	yes
For distribution to	Worldwide
In the following language(s)	Original language of publication
With incidental promotional use	yes

The lifetime unit quantity of new product	Up to 499
Title	System Architecture Optimization Using Hidden Genes Genetic Algorithms with Applications in Space Trajectory Optimization
Instructor name	n/a
Institution name	n/a
Expected presentation date	May 2018
Billing Type	Invoice
Billing Address	Ms. Shadi Darani 112 Quincy St Apt E HANCOCK, MI 49930 United States Attn: Ms. Shadi Darani
Total (may include CCC user fee)	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS

The following terms are individual to this publisher:

Verification of copyright ownership is your responsibility. You should only submit requests for materials that are owned by AIAA. Please review the copyright statement for the source material before submitting a reprint permission request, to ensure that AIAA is the copyright owner:

- For AIAA meeting papers, journal papers, or books with independently authored chapters (e.g., many Progress Series volumes), look at the bottom of the first full-text page (not the cover page). There will be a footnote indicating who holds copyright.
- For other books, look at the copyright statement on the back of the title page.

If the statement reads "Copyright by the American Institute of Aeronautics and Astronautics, Inc.," then AIAA is the copyright owner, and you may submit your request.
If the statement reads otherwise, AIAA does not hold copyright, and cannot grant permission to reprint. You must seek permission from the copyright owner rather than AIAA.
Preferred credit line for reprinted material: From [original title and authors]; reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc. Note that the original source also should be cited in full in the reference list.

Other Terms and Conditions:

Please acknowledge that this portion of your thesis is reprinted by permission of the American Institute of Aeronautics and Astronautics, Inc., and fully cite the conference paper in your reference list.

STANDARD TERMS AND CONDITIONS

1. Description of Service; Defined Terms. This Republication License enables the User to obtain licenses for republication of one or more copyrighted works as described in detail on the relevant Order Confirmation (the “Work(s)”). Copyright Clearance Center, Inc. (“CCC”) grants licenses through the Service on behalf of the rightsholder identified on the Order Confirmation (the “Rightsholder”). “Republication”, as used herein, generally means the inclusion of a Work, in whole or in part, in a new work or works, also as described on the Order Confirmation. “User”, as used herein, means the person or entity making such republication.
2. The terms set forth in the relevant Order Confirmation, and any terms set by the Rightsholder with respect to a particular Work, govern the terms of use of Works in connection with the Service. By using the Service, the person transacting for a republication

license on behalf of the User represents and warrants that he/she/it (a) has been duly authorized by the User to accept, and hereby does accept, all such terms and conditions on behalf of User, and (b) shall inform User of all such terms and conditions. In the event such person is a “freelancer” or other third party independent of User and CCC, such party shall be deemed jointly a “User” for purposes of these terms and conditions. In any event, User shall be deemed to have accepted and agreed to all such terms and conditions if User republishes the Work in any fashion.

3. Scope of License; Limitations and Obligations.

3.1 All Works and all rights therein, including copyright rights, remain the sole and exclusive property of the Rightsholder. The license created by the exchange of an Order Confirmation (and/or any invoice) and payment by User of the full amount set forth on that document includes only those rights expressly set forth in the Order Confirmation and in these terms and conditions, and conveys no other rights in the Work(s) to User. All rights not expressly granted are hereby reserved.

3.2 General Payment Terms: You may pay by credit card or through an account with us payable at the end of the month. If you and we agree that you may establish a standing account with CCC, then the following terms apply: Remit Payment to: Copyright Clearance Center, 29118 Network Place, Chicago, IL 60673-1291. Payments Due: Invoices are payable upon their delivery to you (or upon our notice to you that they are available to you for downloading). After 30 days, outstanding amounts will be subject to a service charge of 1-1/2% per month or, if less, the maximum rate allowed by applicable law. Unless otherwise specifically set forth in the Order Confirmation or in a separate written agreement signed by CCC, invoices are due and payable on “net 30” terms. While User may exercise the rights licensed immediately upon issuance of the Order Confirmation, the license is automatically revoked and is null and void, as if it had never been issued, if complete payment for the license is not received on a timely basis either from User directly or through a payment agent, such as a credit card company.

3.3 Unless otherwise provided in the Order Confirmation, any grant of rights to User (i) is “one-time” (including the editions and product family specified in the license), (ii) is non-exclusive and non-transferable and (iii) is subject to any and all limitations and restrictions (such as, but not limited to, limitations on duration of use or circulation) included in the Order Confirmation or invoice and/or in these terms and conditions. Upon completion of the licensed use, User shall either secure a new permission for further use of the Work(s) or immediately cease any new use of the Work(s) and shall render inaccessible (such as by deleting or by removing or severing links or other locators) any further copies of the Work (except for copies printed on paper in accordance with this license and still in User's stock at the end of such period).

3.4 In the event that the material for which a republication license is sought includes third party materials (such as photographs, illustrations, graphs, inserts and similar materials) which are identified in such material as having been used by permission, User is responsible for identifying, and seeking separate licenses (under this Service or otherwise) for, any of such third party materials; without a separate license, such third party materials may not be used.

3.5 Use of proper copyright notice for a Work is required as a condition of any license granted under the Service. Unless otherwise provided in the Order Confirmation, a proper copyright notice will read substantially as follows: “Republished with permission of [Rightsholder’s name], from [Work's title, author, volume, edition number and year of copyright]; permission conveyed through Copyright Clearance Center, Inc. ” Such notice must be provided in a reasonably legible font size and must be placed either immediately adjacent to the Work as used (for example, as part of a by-line or footnote but not as a separate electronic link) or in the place where substantially all other credits or notices for the new work containing the republished Work are located. Failure to include the required notice results in loss to the Rightsholder and CCC, and the User shall be liable to pay liquidated

damages for each such failure equal to twice the use fee specified in the Order Confirmation, in addition to the use fee itself and any other fees and charges specified.

3.6 User may only make alterations to the Work if and as expressly set forth in the Order Confirmation. No Work may be used in any way that is defamatory, violates the rights of third parties (including such third parties' rights of copyright, privacy, publicity, or other tangible or intangible property), or is otherwise illegal, sexually explicit or obscene. In addition, User may not conjoin a Work with any other material that may result in damage to the reputation of the Rightsholder. User agrees to inform CCC if it becomes aware of any infringement of any rights in a Work and to cooperate with any reasonable request of CCC or the Rightsholder in connection therewith.

4. Indemnity. User hereby indemnifies and agrees to defend the Rightsholder and CCC, and their respective employees and directors, against all claims, liability, damages, costs and expenses, including legal fees and expenses, arising out of any use of a Work beyond the scope of the rights granted herein, or any use of a Work which has been altered in any unauthorized way by User, including claims of defamation or infringement of rights of copyright, publicity, privacy or other tangible or intangible property.

5. Limitation of Liability. UNDER NO CIRCUMSTANCES WILL CCC OR THE RIGHTSHOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES (INCLUDING WITHOUT LIMITATION DAMAGES FOR LOSS OF BUSINESS PROFITS OR INFORMATION, OR FOR BUSINESS INTERRUPTION) ARISING OUT OF THE USE OR INABILITY TO USE A WORK, EVEN IF ONE OF THEM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In any event, the total liability of the Rightsholder and CCC (including their respective employees and directors) shall not exceed the total amount actually paid by User for this license. User assumes full liability for the actions and omissions of its principals, employees, agents, affiliates, successors and assigns.

6. Limited Warranties. THE WORK(S) AND RIGHT(S) ARE PROVIDED “AS IS”. CCC HAS THE RIGHT TO GRANT TO USER THE RIGHTS GRANTED IN THE ORDER CONFIRMATION DOCUMENT. CCC AND THE RIGHTSHOLDER DISCLAIM ALL OTHER WARRANTIES RELATING TO THE WORK(S) AND RIGHT(S), EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ADDITIONAL RIGHTS MAY BE REQUIRED TO USE ILLUSTRATIONS, GRAPHS, PHOTOGRAPHS, ABSTRACTS, INSERTS OR OTHER PORTIONS OF THE WORK (AS OPPOSED TO THE ENTIRE WORK) IN A MANNER CONTEMPLATED BY USER; USER UNDERSTANDS AND AGREES THAT NEITHER CCC NOR THE RIGHTSHOLDER MAY HAVE SUCH ADDITIONAL RIGHTS TO GRANT.

7. Effect of Breach. Any failure by User to pay any amount when due, or any use by User of a Work beyond the scope of the license set forth in the Order Confirmation and/or these terms and conditions, shall be a material breach of the license created by the Order Confirmation and these terms and conditions. Any breach not cured within 30 days of written notice thereof shall result in immediate termination of such license without further notice. Any unauthorized (but licensable) use of a Work that is terminated immediately upon notice thereof may be liquidated by payment of the Rightsholder's ordinary license price therefor; any unauthorized (and unlicensable) use that is not terminated immediately for any reason (including, for example, because materials containing the Work cannot reasonably be recalled) will be subject to all remedies available at law or in equity, but in no event to a payment of less than three times the Rightsholder's ordinary license price for the most closely analogous licensable use plus Rightsholder's and/or CCC's costs and expenses incurred in collecting such payment.

8. **Miscellaneous.**

8.1 User acknowledges that CCC may, from time to time, make changes or additions to the Service or to these terms and conditions, and CCC reserves the right to send notice to the User by electronic mail or otherwise for the purposes of notifying User of such changes or

additions; provided that any such changes or additions shall not apply to permissions already secured and paid for.

8.2 Use of User-related information collected through the Service is governed by CCC's privacy policy, available online here:
http://www.copyright.com/content/cc3/en/tools/footer/privacypolicy.html.

8.3 The licensing transaction described in the Order Confirmation is personal to User. Therefore, User may not assign or transfer to any other person (whether a natural person or an organization of any kind) the license created by the Order Confirmation and these terms and conditions or any rights granted hereunder; provided, however, that User may assign such license in its entirety on written notice to CCC in the event of a transfer of all or substantially all of User's rights in the new material which includes the Work(s) licensed under this Service.

8.4 No amendment or waiver of any terms is binding unless set forth in writing and signed by the parties. The Rightsholder and CCC hereby object to any terms contained in any writing prepared by the User or its principals, employees, agents or affiliates and purporting to govern or otherwise relate to the licensing transaction described in the Order Confirmation, which terms are in any way inconsistent with any terms set forth in the Order Confirmation and/or in these terms and conditions or CCC's standard operating procedures, whether such writing is prepared prior to, simultaneously with or subsequent to the Order Confirmation, and whether such writing appears on a copy of the Order Confirmation or in a separate instrument.

8.5 The licensing transaction described in the Order Confirmation document shall be governed by and construed under the law of the State of New York, USA, without regard to the principles thereof of conflicts of law. Any case, controversy, suit, action, or proceeding arising out of, in connection with, or related to such licensing transaction shall be brought, at CCC's sole discretion, in any federal or state court located in the County of New York, State of New York, USA, or in any federal or state court whose geographical jurisdiction covers the location of the Rightsholder set forth in the Order Confirmation. The parties expressly submit to the personal jurisdiction and venue of each such federal or state court.If you have any comments or questions about the Service or Copyright Clearance Center, please contact us at 978-750-8400 or send an e-mail to info@copyright.com.

v 1.1

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Shadi Darani <sahmadid@mtu.edu> 2:54 PM (8 minutes ago)

to Ossama

Dear Dr. Abdelkhalik,

I am writing my PhD dissertation and I would like to reuse some of our publications during the past few years. Since you have the copyright permission of these papers, I was wondering if you would agree to reusing them in my dissertation.

Convergence Analysis of Hidden Genes Genetic Algorithms in Space Trajectory Optimization, Journal of Aerospace Information Systems
https://arc.aiaa.org/doi/abs/10.2514/1.I010564

Space Trajectory Optimization Using Hidden Genes Genetic Algorithms, Journal of Spacecraft and Rockets
https://arc.aiaa.org/doi/abs/10.2514/1.A33994

Evolving Hidden Genes in Genetic Algorithms For Systems Architecture Optimization, Journal of Dynamic Systems, Measurement, and Control

Thank you

...

Ossama Abdelkhalik 2:59 PM (3 minutes ago)

to me

Shadi,
Yes you can use these two papers in your dissertation.
Ossama