Convergence Analysis of Hidden Genes Genetic Algorithms in Space Trajectory Optimization

Shadi A. Darani[1] and Ossama Abdelkhalik[2]
Michigan Technological University, Houghton, Michigan 49931-1295.

I. Background and Introduction

A systems architecture optimization problem is characterized by being a variable-size design space (VSDS) problem; i.e. the number of variables is variable. To see that, consider a space mission from Earth to Jupiter. Consider a solution to the trajectory optimization problem that has two fly-bys and another solution that has three fly-bys. While the two solutions are for the same optimization problem (same objective function), the number of variables in one solution is different from the number of variables in the other solution (adding a fly-by implies adding some variables such as the fly-by hight and the fly-by plane). Hence, the design space dimension (size) varies among different solutions. There are different deterministic, heuristic, and hybrid algorithms proposed for the interplanetary trajectory optimization problems. References [1, 2] use deterministic algorithms based on grid or tree search. Heuristic algorithms include genetic algorithms (GAs), differential evolution, particle swarm optimization, adaptive simulated annealing [3–6]. Reference [7–9] use hybrid methods like Multistart, monotonic basin hopping algorithm, and machine learning. In some of these algorithms either a pruning method is applied to limit the possible mission structures, or the solution structure (fly-by sequence) is assumed known a priori. Recent studies on variations of GAs, including hidden genes genetic algorithms (HGGAs), have been able to solve trajectory optimization problems in the general form with unknown fly-bys and DSMS.

In standard GAs, the variables of the optimization problem are coded as genes; each solution

[1] Corresponding author, PhD Candidate, Department of Mechanical Engineering-Engineering Mechanics, 815 R.L.Smith Bldg, 1400 Townsend Dr., sahmadid@mtu.edu
[2] Associate Professor, Department of Mechanical Engineering-Engineering Mechanics, 815 R.L. Smith Bldg, 1400 Townsend Dr, Houghton, MI 49931, USA; ooabdelk@mtu.edu; Member AIAA

is a chromosome. In **Figure** 1, a solution with l variables is shown as a chromosome with l genes $g_1, g_2, \ldots, g_l$. The genetic operations of selection, mutation, and crossover are applied on a population of these chromosomes. Through generations (iterations), theses populations evolve toward the optimal solution. All the GA operations are defined on fixed-length chromosomes; and hence the standard GAs can only handle problems of Fixed-Size Design Space (FSDS).

Fig. 1 Solutions are represented as chromosomes (string of genes) in standard GA.

The biologically inspired concept of hidden genes was recently proposed in evolutionary algorithms and hidden genes genetic algorithms were introduced to model this type of VSDS optimization problems. **Figure** 2 shows an illustration for a chromosome in HGGA. A binary tag is appended to each gene. This binary tag determines whether the gene is hidden or active. When a gene is hidden, it does not get transcribed during the objective function evaluation. In other words, although the hidden gene exists in the chromosome, it does not affect the fitness (or objective function value) of the solution. It does carry information to the next generation though. This concept of hidden genes enables solutions of different lengths to be coded in chromosomes of equal lengths. For instance, in a space trajectory optimization problem, a solution that has two fly-bys would have the same chromosome length as that of a solution that has three fly-bys, the only difference would be in the number of hidden genes in each chromosome.

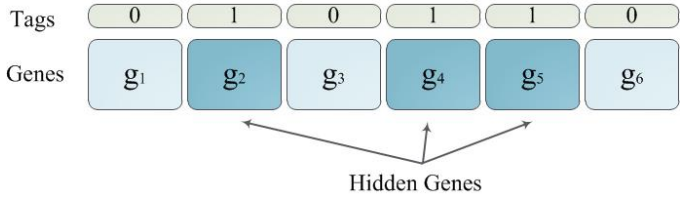

Fig. 2 Tags and genes in a solution of HGGA.

The HGGAs were implemented to optimize interplanetary space trajectories where it demonstrated the capability of searching for the optimal number of fly-by maneuvers needed for the mission, along with the fly-by planets, the number of deep space maneuvers and their direction and magnitude, in addition to the rest of the trajectory design variables [10, 11].

In the evolution process, chromosomes (solutions) evolve over subsequent iterations generating new solutions. 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 are introduced in [10]. In one mechanism, the tags evolve through stochastic operations, while in another one the tags evolve through logical operations. The performance of these mechanisms is tested on different VSDS problems, including space trajectory optimization problems. Although the HGGAs converge toward optimal solutions and find best known solutions for benchmark problems as reported in [10], there is no analytical proof that the HGGAs, with the tags evolution mechanisms, are convergent. In [12], 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 [13] 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 [14, 15]. Hence, with the introduction of the new evolution mechanisms, a more comprehensive investigation of the HGGAs properties and convergence characteristics is needed.

This paper 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 [10] 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 $1/f_u > 0$, where f_u is the fuel consumption. Hence, the problem can be treated as a maximization problem. In section I A, a review for the Markov model for binary canonical genetic algorithms (CGAs) is presented and its convergence is analyzed. Section II briefs the tag evolution mechanisms. In section III, the transition matrix of different HGGA mechanisms are derived and the convergence conditions are investigated. Section IV presents numerical tests carried out on a space trajectory optimization problem.

A. 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 [16]. Several studies have investigated the convergence behavior of the GA explicitly using the Markov chain analysis [16–20]. The minimum conditions for convergence of EGAs in the realm of Markov chain model can be found in details in [16, 20, 21]. 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. The materials of this section are a nearly verbatim adaptation of works done by Rudolph [20] and Davis [16]. 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:

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

  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. Let l be the chromosome length, $M = 2^l$ be the constant population size, and $m = 2^{nl}$. 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 [21]. 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 [20].

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 [20].

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

$$\mathbf{P}^\infty = \lim_{k\to\infty} \mathbf{P}^k = \lim_{k\to\infty} \begin{bmatrix} \mathbf{D}^k & 0 \\ \sum_{i=0}^{k-1} \mathbf{T}^i\mathbf{R}\mathbf{D}^{k-i} & T^k \end{bmatrix} = \begin{bmatrix} \mathbf{D}^\infty & 0 \\ \mathbf{R}_\infty & 0 \end{bmatrix} \tag{2}$$

is a stable stochastic matrix with $\mathbf{P}^\infty = l'\mathbf{p}^\infty$, 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 $m < i \leq n$ [20].

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 [20].

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 [20].

Theorem 4: The CGA with parameter ranges as in Theorem 3 does not converge to the global

optimum [20].

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 [20].

Theorem 6: The canonical GA as in Theorem 3 maintaining the best solution found over time after selection converges to the global optimum [20].

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 [20]:

$$\mathbf{C}^+ = \begin{bmatrix} \mathbf{C} & & & \\ & \mathbf{C} & & \\ & & \ldots & \\ & & & \mathbf{C} \end{bmatrix}, \mathbf{M}^+ = \begin{bmatrix} \mathbf{M} & & & \\ & \mathbf{M} & & \\ & & \ldots & \\ & & & \mathbf{M} \end{bmatrix}, \mathbf{S}^+ = \begin{bmatrix} \mathbf{S} & & & \\ & \mathbf{S} & & \\ & & \ldots & \\ & & & \mathbf{S} \end{bmatrix} \tag{3}$$

Then we can write:

$$\mathbf{C}^+\mathbf{M}^+\mathbf{S}^+ = \begin{bmatrix} \mathbf{CMS} & & & \\ & \mathbf{CMS} & & \\ & & \ldots & \\ & & & \mathbf{CMS} \end{bmatrix} \tag{4}$$

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 $f(\pi_0(i)) < b$, otherwise $u_{ii} = 1$. Therefore, there is one entry in each row and for every state j with $f(\pi_0(j)) < max[f(\pi_k(j))|k = 1 \dots n]$, the elements will be $u_{ij} = 0$ for all is. Hence, the structure of the upgrade matrix can be written as [20]:

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_{11} & & & \\ \mathbf{U}_{21} & \mathbf{U}_{22} & & \\ \dots & \dots & \dots & \\ \mathbf{U}_{2^l,1} & \mathbf{U}_{2^l,2} & \dots & \mathbf{U}_{2^l,2^l} \end{bmatrix} \tag{5}$$

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:

$$\mathbf{P}^+ = \begin{bmatrix} \mathbf{P} & & & \\ & \mathbf{P} & & \\ & & \dots & \\ & & & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & & & \\ \mathbf{U}_{21} & \mathbf{U}_{22} & & \\ \dots & \dots & \dots & \\ \mathbf{U}_{2^l,1} & \mathbf{U}_{2^l,2} & \dots & \mathbf{U}_{2^l,2^l} \end{bmatrix} = \begin{bmatrix} \mathbf{PU}_{11} & & & \\ \mathbf{PU}_{21} & \mathbf{PU}_{22} & & \\ \dots & \dots & \dots & \\ \mathbf{PU}_{2^l,1} & \mathbf{PU}_{2^l,2} & \dots & \mathbf{PU}_{2^l,2^l} \end{bmatrix} \tag{6}$$

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 Eq. (6) to Eq. (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 Eq. (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 Eq. (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 III, these matrices are explicitly derived and it is shown that in the HGGA, the **C**, **M**, and **S** are stochastic, positive, and column-allowable.

II. Tags Evolution Mechanisms in HGGA

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. There are 12 different mechanisms for tags evolution that will be investigated in this paper. The complete explanation of these mechanisms can be found in [10]. Here, a brief description of each mechanism is provided. 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%.

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, 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.

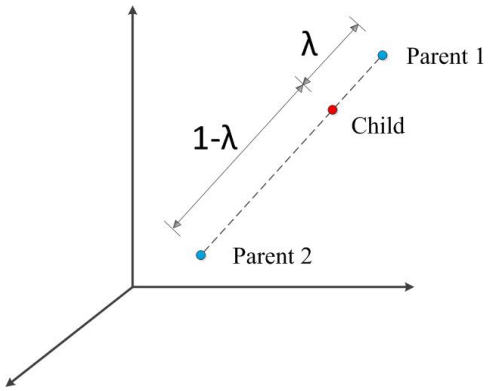

Fig. 3 Representation of arithmetic crossover in $\mathbb{R}^3$.

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 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: two tags are assigned for each gene, one recessive and one dominant. First, the mutation operation is 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.

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 in the generation.

III. Markov Chain Model of Hidden Genes Genetic Algorithm

The HGGA using any of the stochastic or logical mechanisms, defined in section II, is 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 Eq. (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.

A. 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 [16]:

$$P_1(i|\bar{n}) = \frac{n(i).R(i)}{\sum_{j \in S} n(j).R(j)} \tag{7}$$

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 [16]:

$$P_1(\bar{m}|\bar{n}) = \binom{M}{\bar{m}} \prod_{i \in S} P_1(i|\bar{n})^{m(i)} \tag{8}$$

where,

$$\binom{M}{\bar{m}} = \frac{M!}{\prod_{i \in S} (m(i)!)} \tag{9}$$

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.

B. 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 a nonzero value for the mutation probability, 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 [16]:

$$P_2(i|\bar{n}) = \sum_{j \in S} p_m^{H(i,j)}(1-p_m)^{L-H(i,j)} P_1(j|\bar{n}) = \frac{1}{(1+\alpha)^L} \sum_{j \in S} \alpha^{H(i,j)} P_1(j|\bar{n}), \bar{n} \in S', i \in S \tag{10}$$

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

$$\therefore P_2(i|\bar{n}) = \frac{\sum_{j\in S} \alpha^{H(i,j)}(n(j).R(j))}{(1+\alpha)^L.\sum_{k\in S} n(k).R(k)} \tag{11}$$

The multinomial distribution for $P_2(\bar{m}|\bar{n})$ can be defined as [16]:

$$P_2(\bar{m}|\bar{n}) = \binom{M}{\bar{m}} \prod_{i\in S} P_2(i|\bar{n})^{m(i)} \tag{12}$$

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 Eq. (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 [16]:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S}\left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L-1}\sum_s I(i,j,k,s)\right) + (1-p_c).P_1(k|\bar{n}) \tag{13}$$

Therefore the conditional probability of producing k via selection, mutation, and crossover operations is [16]:

$$P_3(i|\bar{n}) = \frac{1}{(1+\alpha)^L}\sum_{j\in S} \alpha^{H(i,j)} P_2'(j|\bar{n}) \tag{14}$$

Then:

$$P_3(\bar{m}|\bar{n}) = \binom{M}{\bar{m}}.\prod_{i\in S} P_3(i|\bar{n})^{m(i)} \tag{15}$$

By inspection of Eq. (13) and Eq. (14), it can be seen that this three-operator Markov chain is primitive. Then, based on the results of section I A 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:

$$P_2(i|\bar{n}) = \sum_{j \in S} p_m^{H(i,j)} (1-p_m)^{L-H(i,j)} p_{mt}^{H_t(i,j)} (1-p_{mt})^{L_t - H_t(i,j)} P_1(j|\bar{n}) \tag{16}$$

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:

$$P_2'(k|\bar{n}) = \sum_{i \in S} \sum_{j \in S} \left(P_1(i|\bar{n}) P_1(j|\bar{n}) \frac{p_c}{L-1} \frac{1}{L_t - 1} \sum_s I'(i,j,k,s,s_t) \right) + (1-p_c) P_1(k|\bar{n}) \tag{17}$$

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:

$$P_3(i|\bar{n}) = \frac{1}{(1+\alpha)^{L+L_t}} \sum_{j \in S} \alpha^{H(i,j)} P_2'(j|\bar{n}) \tag{18}$$

Then the transition matrix for Mechanism A can be computed by substituting Eq. (18) into Eq. (15). Note that L is replaced by $L + L_t$ to account for the additional tags. By inspection

of Eq. (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 Eq. (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 Eq. (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 Eqs. (14) and (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 II. The selection and crossover transition probability is defined as follows:

$$P_2'(k|\bar{n}) = \sum_{i \in S} \sum_{j \in S} P_1(i|\bar{n}) P_1(j|\bar{n}) p_c F_A(i,j,k,\lambda) F_{T_1}(i,j,k, f_{m_1}(i), f_{m_1}(j)) + (1 - p_c) P_1(k|\bar{n}) \tag{19}$$

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 II, 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:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n})P_1(j|\bar{n})p_c F_A(i,j,k,\lambda)F_{T_2}(i,j,k,f_{m_2}(i),f_{m_2}(j)) + (1-p_c)P_1(k|\bar{n}) \tag{20}$$

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:

$$P_2(i|\bar{n}) = \sum_{j\in S} p_m^{H(i,j)}(1-p_m)^{L-H(i,j)} p_{mt}^{H_t(i,j)}(1-p_{mt})^{L_t-H_t(i,j)} P_1(j|\bar{n}) \tag{21}$$

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:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S}\left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L-1}\sum_s I(i,j,k,s)\right) + (1-p_c).P_1(k|\bar{n}) \tag{22}$$

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:

$$P_2(i|\bar{n}) = \sum_{j\in S} p_m^{H(i,j)+H_t(i,j)}(1-p_m)^{L+L_t-H(i,j)-H_t(i,j)} P_1(j|\bar{n}) \tag{23}$$

which results in a positive and stochastic mutation matrix. Also the stochastic crossover

transition probability can be calculated as follows:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S}\left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L+L_t-1}\sum_s I(i,j,k,s)\right) + (1-p_c).P_1(k|\bar{n}) \quad (24)$$

- 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:

$$P_2(i|\bar{n}) = \sum_{j\in S} p_m^{H(i,j)}(1-p_m)^{L+L_t-H(i,j)}P_1(j|\bar{n}) \quad (25)$$

  which results in a positive and stochastic mutation matrix. The stochastic crossover probability matrix is similar to Eq. (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 Eq. (23) which is stochastic and positive. The crossover probability matrix is similar to Eq. (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:

$$P_2(i|\bar{n}) = \sum_{j\in S} p_m^{H(i,j)+H_a(i,j)}(1-p_m)^{L+2L_t-H(i,j)-H_a(i,j)}P_1(j|\bar{n}) \quad (26)$$

  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:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S}\left(P_1(i|\bar{n}).P_1(j|\bar{n}).\frac{p_c}{L-1}.\frac{1}{L_t-1}\sum_s I'(i,j,k,s,s_t)\right) + (1-p_c).P_1(k|\bar{n}) \tag{27}$$

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 Eq. (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 Eq. (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:

$$P_2'(k|\bar{n}_1) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}_1).P_1(j|\bar{n}_1).F_{HO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}_1) \tag{28}$$

and for the second half of the children:

$$P_2''(k|\bar{n}_2) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}_2).P_1(j|\bar{n}_2).F_{AO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}_2) \tag{29}$$

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 Eq. (25). The crossover probability matrix is:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}).P_1(j|\bar{n}).F_{HO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}) \quad (30)$$

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 Eq. (25). The crossover probability matrix is:

$$P_2'(k|\bar{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\bar{n}).P_1(j|\bar{n}).F_{AO}(i,j,k).\frac{p_c}{L-1}\sum_s I(i,j,k,s) + (1-p_c).P_1(k|\bar{n}) \quad (31)$$

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.

IV. Statistical Analysis

The results of the previous section show that the HGGAs using any of the proposed tags evolution mechanisms are convergent. Here, some mechanisms are tested on an interplanetary trajectory problem and their numerical convergence is investigated. The problem of interplanetary trajectory optimization is defined as finding the minimum cost trajectory of a spacecraft, traveling

from one celestial body to another. The cost of the mission is the spacecraft fuel consumption (f_u) and the objective function is to maximize $1/f_u$. The trajectory is determined through finding the position and velocity vectors of the spacecraft at any moment. The spacecraft can have multiple fly-bys and/or deep-space maneuvers (DSMs). This type of problem is a VSDS system architecture optimization problem, where the number of variables can change based on the topology of the solution. Several global trajectory optimization methods have been proposed to solve the mutli-gravity assist deep-space maneuvers (MGADSM) problem.

It has been demonstrated that the HGGAs can search for optimal solution architectures, and find the optimal topology for bench mark interplanetary trajectory optimization problems [10, 12].

In this section, the Earth-to-Jupiter and Earth-to-Mars space missions are selected to numerically investigate the convergence of Mechanisms A, B, C, and D. Note that the variables for both of the problems are considered continuous in this section.

The Earth-to-Jupiter trajectory optimization problem is defined as finding the optimal fuel consumption trajectory of a spacecraft traveling from Earth to Jupiter, constrained to a maximum of two fly-bys around any two planets in the solar system, and a maximum of two DSMs in each leg (a leg is a trajectory segment between two successive fly-bys). The launch time must be between $1-30$ September 2016 and the arrival time must be between 1 September to 31 December 2021. The time of flight for each leg can be between 80 and 800 days. The problem variables and their upper and lower bounds are shown in Table 1. The variables that can be hidden are the fly-by planets and the DSMs in each leg. Each planet in the solar system is given a number as indicator; with Mercury as 1 and Neptune as 8. The The maximum number of fly-bys are two, hence two genes are assigned for this variable. Each gene can have values between 1 and 8. For example, if the genes are 2 and 6, it means that the first fly-by is around Venus and the second fly-by is around Saturn. A tag is assigned to each gene that determines if it is active or hidden. If a tag is 0, the corresponding fly-by is active and if it is 1, that fly-by is hidden. For the same example with fly-bys around Venus and Saturn, if the tag of the first fly-by is 0 and the tag of the second fly-by is 1 it means that the solution has only one fly-by around Venus. The genes go through selection, mutation, and crossover operations and the tags go through the evolutionary mechanisms defined in Section II. Similarly,

since there are maximum three legs and maximum two DSMs in each leg, there are six genes for the DSMs variables. Each gene has a tag that determines if the DSM is active or not. Keep in mind that there are some dependent variables in the problem. For example, if a DSM is hidden, the corresponding maneuver time, magnitude, and direction are hidden too. Although there is no need to define new tags for them as their status can be determined by the tags of the DSM.

Table 1 Lower and upper bounds of Earth-to-Jupiter problem

Design Variable	Lower Bound	Upper Bound
Number of fly-bys	0	2
Fly-by planet	1 ($Mercury$)	8 ($Neptune$)
Number of DSMs in each mission leg	0	2
Flight Direction	Posigrade	Retrograde
Departure Date	01 Sep.2016	30 Sep.2016
Arrival Date	01 Sep.2021	31 Dec.2021
Time Of Flight (TOF) (days)	80	800
Fly-by normalized pericenter altitude	0.1	10
Fly-by plane rotation angle (rad)	0	2π
Epoch of DSM	0.1	0.9
DSM (km/s)	-5	5

In the HGGAs simulations, the number of populations is 200 and the number of generations is 200. The problem is solved five times and results show that the mechanisms are convergent. The objective value versus number of generations is shown in **Figure** 4 for Mechanism C. Other mechanisms have similar trends as Mechanism C. A sample trajectory of Earth-to-Jupiter using Mechanism A is shown in **Figure** 5. In this figure, the cost of Earth-to-Jupiter mission is 10.1985 km/s.

Moreover, the same problem is solved 100 times (100 identical numerical experiments) for each of the mechanisms A, B, C, and D, and the success rate of each of the mechanisms is assessed based on these 100 runs. The success rate of 0.2 means that out of 100 runs, it is expected that 20 runs generate a solution within an error of 5% of the best solution found overall.

As shown in the **Figure** 6, the success rates also converge, although different mechanisms have different success rates. Using Mechanism D, the success rate is the highest and is around 55%. Mechanism C has the lowest success rate of 20%. Keep in mind that the performance of the mechanisms can not be compared based on only the success rate, since a mechanism with high success rate might have a poor average objective value. The figures of this section are only provided

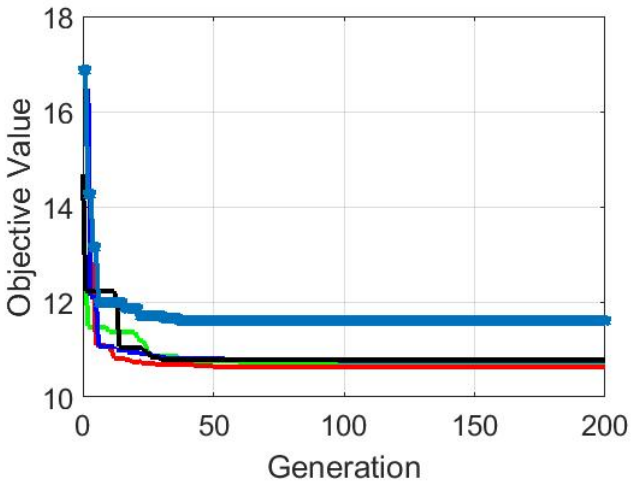

Fig. 4 Convergence of 5 runs for mechanism C in Earth-to-Jupiter mission.

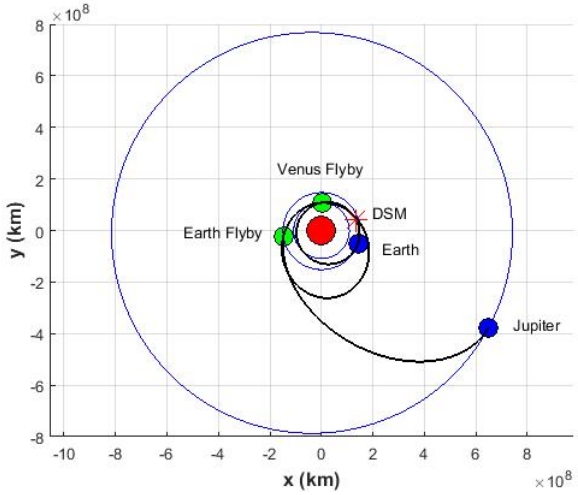

Fig. 5 Earth-to-Jupiter trajectory using Mechanism A

for convergence investigation of the mechanisms.

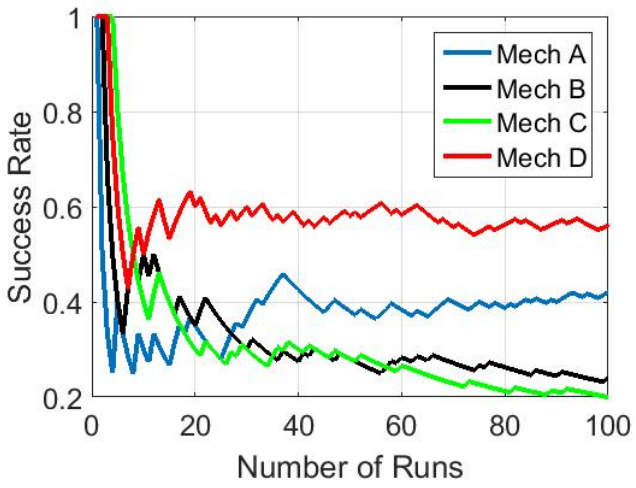

Fig. 6 Success rate of different mechanisms in Earth-to-Jupiter mission.

Conclusions

The HGGAs are designed to handle the optimization of variable-sized design space problems, where there is one or two tags assigned to each gene to determine its active/hidden status. The genes evolve through selection, mutation, and crossover operators, while the tags can evolve through several evolution mechanisms designed specifically for them. In this paper, the convergence of different HGGA tags evolution mechanisms are investigated. The HGGAs are modeled as homogeneous finite Markov processes where selection, mutation, and crossover matrices are derived for each of the mechanisms. It is shown that if the mutation probability is non-zero and the algorithm keeps the best solution found over time, the investigated HGGA mechanisms converge toward the global optimum. The convergence of some of the mechanisms is demonstrated numerically in this paper on a space trajectory design optimization problem and the success rate is assessed numerically for some of the algorithms.

Acknowledgments

This work was funded by National Science Foundation, award # 1446622.

References

[1] Vose, M. D. and Liepins, G. E., "Hodographic-Shaping Method for Low-Thrust Interplanetary Trajectory Design," *AIAA Journal of Spacecraft and Rockets*, Vol. 52, No. 3, 2015, pp. 728âĂŞ–738.

[2] Olympio, J. T. and Yam, C. H., "Deterministic Method for Space Trajectory Design with Mission Margin Constraints," *61st International Astronautical Congress*, Prague, CZ, 2010.

[3] Vasile, M., "A Global Approach to Optimal Space Trajectory Design," *In Proceedings of the 2003 AAS/AIAA Space Flight Mechanics Meeting*, AAS/AIAA, Ponce, Puerto Rico, 2003, pp. 629–648.

[4] Vasile, M. and de Pascale, P., "Preliminary design of multiple gravity-assist trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 4, 2006, pp. 794âĂŞ805.

[5] Myatt D. R., Becerra V. M., N. S. J. and M., B. J., "Advanced global optimisation for mission analysis and design," *European Space Agency, Ariadna Final Report*, , No. 03-4101a, 2004.

[6] Izzo D., Becerra V. M., M. D. R. N. S. J. and M., B. J., "Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories," *Journal of Global Optimization*, Vol. 38, 2007, pp. 283–296.

[7] Englander, J. A. and Englander, A. C., "Tuning Monotonic Basin Hopping: Improving the Efficiency of Stochastic Search as Applied to Low-Thrust Trajectory Optimization," *24th International Symposium on Space Flight Dynamics*, Laurel, MD, 2014.

[8] Addis B., Cassioli A., L. M. and F., S., "A global optimization method for the design of space trajectories," *Computational Optimization and Applications*, Vol. 48, 2011, pp. 635âĂŞ–652.

[9] Cassioli A., Di Lorenzo D., L. M. S. F. and M., S., "Machine learning for global optimization," *Computational Optimization and Applications*, Vol. 51, 2012, pp. 279–303.

[10] Abdelkhalik, O. and Darani, S. A., "Hidden Genes Genetic Algorithms for Systems Architecture Optimization," *In Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, Denver, Colorado, 2016, pp. 629–636.

[11] Darani, S. A. and Abdelkhalik, O., "Developments on The Optimization of Interplanetary Trajectories using Hidden Genes Genetic Algorithms," *In Proceedings of the 2016 AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum*, 2016.

[12] Abdelkhalik, O., "Hidden Genes Genetic Optimization for Variable-Size Design Space Problems," *Springer, Journal of Optimization Theory and Applications*, Vol. 156, No. 2, 2012, pp. 450–468.

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

[14] Muhlenbein, H., "Evolutionary Algorithms: Theory and Application," *Local Search in Combinatorial Optimization*, 1993.

[15] Muhlenbein, H., "Evolution in Time and Space, the parallel Genetic Algorithm," *Foundation of the Genetic Algorithm*, 1991, pp. 316–337.

[16] Davis, T. E. and Principe, J. C., "A Markov Chain Framework for the Simple Genetic Algorithm," *Evolutionary Computation*, Vol. 1, 1993, pp. 269–288.

[17] Vose, M. D. and Liepins, G. E., "Punctuated Equilibria in Genetic Search," *Complex Systems*, Vol. 5, 1991, pp. 31–44.

[18] Nix, A. E. and Vose, M. D., "Modeling Genetic Algorithms with Markov Chains," *Annals of Mathematics and Artificial Intelligence*, Vol. 5, 1992, pp. 79–88.

[19] Suzuki, J., "A Markov Chain Analysis on a Genetic Algorithm," *In Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 146–153.

[20] Rudolph, G., "Convergence Analysis of Canonical Genetic Algorithm," *IEEE Transaction on Neural Networkss*, Vol. 5, No. 1, 1994, pp. 96–101.

[21] Agapie, A., "Genetic algorithms: Minimal conditions for convergence," *In Proceedings of the 2005*

European Conference on Artificial Evolution, Heidelberg, Berlin, Germany, 2005, pp. 181–193.