Hidden Genes Genetic Algorithms for Systems Architecture Optimization

Ossama Abdelkhalik
Michigan Technology University
1400 Townsend Drive
Houghton, MI, 49931
ooabdelk@mtu.edu

Shadi Darani
Michigan Technology University
1400 Townsend Drive
Houghton, MI, 49931
sahmadid@mtu.edu

ABSTRACT

The concept of hidden genes was recently introduced in genetic algorithms to handle variable-size design space optimization problems. This paper presents new developments in hidden genes genetic algorithms. Mechanisms for assigning (selecting) the hidden genes in the chromosomes of genetic algorithms are presented. In the proposed mechanisms, a tag is assigned for each gene; this tag determines whether the gene is hidden or not, while they evolve over generations using stochastic operations. These mechanisms are tested on mathematical optimization problems and on a trajectory optimization problem for a space mission to Jupiter. In the conducted tests, one of the proposed hidden genes assignment mechanism has enabled the hidden genes genetic algorithms to find better (lower cost) solutions, while other mechanisms has shown to be able to find close solutions.

Keywords

genetic algorithms; hidden genes; architecture optimization; topology optimization

1. INTRODUCTION

Global optimization techniques have been studied for different problems. One of these problems is system architecture optimization in which not only the desired objective, but also the system architecture (topology) is optimized. These problems are called variable-sized design space (VSDS) problems. The design space and number of variables depend on the system structure. Automated construction and smart buildings, electric grids, and medical technology are some of the fields that the importance of system architecture arises. Generally speaking, the optimizing techniques are divided into local and global optimization techniques. Gradiend-based methods [3] are one example of local optimizing techniques. Global search methods include genetic algorithms (GAs) [7], particle swarm optimization [10], ant colony optimization [4], and differential evolution [12]. Most

practical optimization problems are usually solved using local minima algorithms and hence there is an urge for global search algorithms in optimizing the system architecture. In the other hand, most existing optimization algorithms are designed to handle only Fixed-Size Design Space (FSDS) problems, and hence they cannot be used for system architecture optimization (optimize the system variables, given a known system architecture). The few algorithms that can be used in system architecture optimization are problem-specific. Examples include truss and frame structure optimization [2, 8] and topology optimization of nanophotonic devices [15].

The problem of VSDS can be formulated as follows:

$$\text{Minimize} \quad f(\vec{x}, N)$$
$$\text{Subject to} \quad \vec{x}^l \le \vec{x} \le \vec{x}^u \qquad (1)$$
$$\text{Where} \quad \vec{x} = [x_1, x_2, ..., x_N]^T,$$

where N is the number of design variables, and $\vec{x}^u$ and $\vec{x}^l$ are the upper and lower bounds of the variables $\vec{x}$, respectively. The number of variables (which is a variable) depends on the architecture of the solution.

Inspired by the concept of hidden genes in biology, this project investigates revolutionary optimization algorithms that can optimize among different solution topologies and autonomously develop new topologies. **Sections** 1.1 and 1.2 present necessary background material.

1.1 Genetic Algorithm

In standard GAs, the mechanics of natural selection and genetics [7] are simulated. For each solution a chromosome is considered which is a set of coded variables called genes. The algorithm starts by applying the genetic operations of crossover and mutation on a population of these chromosomes. Through generations (iterations), this population converges to nearly-optimal solutions. **Figure** 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 1: In standard GA, a chromosome (code) is a string of genes that represent a solution

In each generation of GA, a number of operations are ap-

plied to the current population, including selection, crossover, and mutation. In selection operation, the chromosomes that are more fit, have the higher probability of being selected as parents. After parents are selected, the operations of crossover and mutation are applied on them to create new children chromosomes. For example in single point crossover, a random point in parents strings is selected and the gene strings of both sides of that point are swaped 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 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 selecton, crossover, and mutation in each generation, the population converges to near optimal solution.

1.2 Hidden Genes Genetic Algorithm

The deoxyribonucleic acid (DNA) in genetics is organized into a long structure called chromosome (**Figure** 2). Genes, which are instructions for making a protein, are contained in the DNA and are coded with a specific language with 64 words. Difference in each word, makes genes to produce different proteins, making cells of distinct organs function differently. For example, an eye cell is shut off in the lung and breathing genes are shut off in the eyes. This duty scheduling is done in another layer of coding which tells genes in a cell what coding they should read and what they should shut off [13]. The genes that are shut off are called hidden genes.
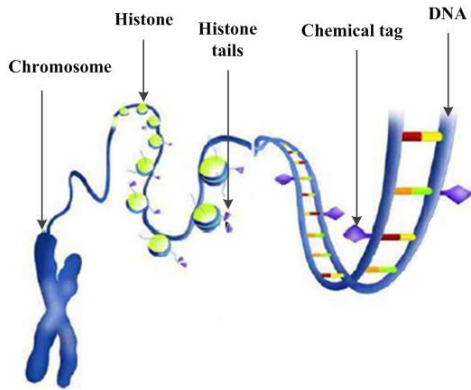

Figure 2: Chemical tags (purple diamonds) and the "tails" of histone proteins (purple triangles) mark DNA to determine which genes will be transcribed. (picture is modified from) [14]

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. Consider two chromosome with
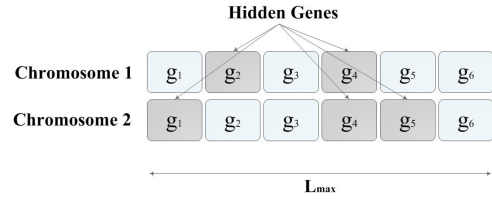

Figure 3: Hidden genes and effective genes in two different chromosomes [6]

different lengths. Assume that there are five genes in the first chromosome and three genes in the second chromosome (represented by binary bits in **Figure** 3). Also assume that the maximum number of genes (variables) in a chromosome is seven. To make this problem a FSDS problem, two hidden genes are added to the first chromosome, and four 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). This algorithm is called Hidden Genes Genetic Algorithm (HGGA).

A simple example of a single-point crossover operator in HGGA is shown in **Figure** 4. In this figure, the crossover point is between genes 2 and 3. After the genes are swaped, 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 explained in next section.

2. HIDDEN GENES ASSIGNMENT METHODS

In this paper the mechanism of assigning hidden genes in a chromosome is addressed. A primitive mechanism (called "feasibility mechanism") is introduced in previous work of HGGA [1]. In the feasibility mechanism, the genes would be hidden one by one from one end of the string until the chromosome gives a feasible solution regarding constraints.

As discussed in Section 1.2, the protein of each gene makes it to be read or hidden. This gave us the idea to use a tag for each gene to make it hidden or active. to code such tags, binary digits of 0 and 1 are assigned to each tag, as shown in **Figure** 5. 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).

The tags values are evolved through crossover and/or mutation operations, in a similar way to that of the design variables. Four mechanisms are studied using this concept. These four mechanisms are:

1. Mechanism A: tags can crossover independently from the genes. In other words, the tag of gene X can
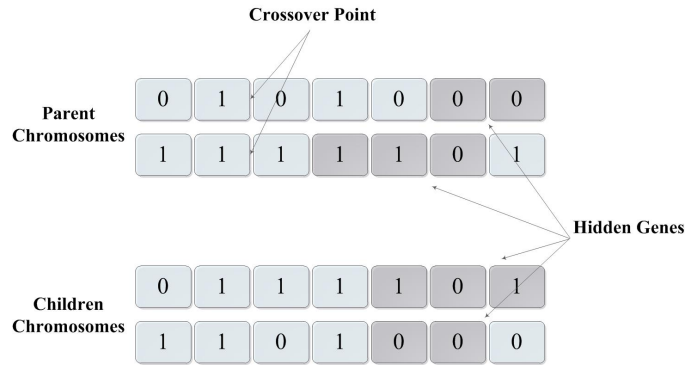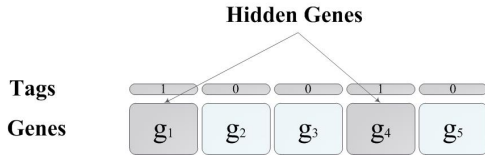
Figure 4: Crossover operation in HGGA [6]

Figure 5: HGGA and the tags concept

change its position with tag of gene Y while the genes remains same themselves. This mechanism can be interpreted as a $2-D$ multiple crossover operator, one direction through tags and one direction through genes. Before applying the crossover operator, tags undergo a mutation with probability of 10%.

2. Mechanism B: a logical tag-based fitness guided (arithmetic) crossover in which two intermediate chromosomes are produced. In these intermediate chromosomes, the genes are produced from a single crossover operator on parents and the tags are the outcome of the Active OR logic on parents's tags. In other words, parent X will have intermediate offspring Xx, and parent Y will have intermediate offspring Yy as the criterion of the arithmetic crossover. The actual offspring is then created by a fitness guided crossover operator on the parents and is closer to the parent with lower cost for its intermediate offspring.

3. Mechanism C: the arithmetic crossover is used with a modified cost function based on the number of genes that are hidden. The offspring is biased toward better parent (lower cost) with more hidden genes, which is coded as a modified cost function.

4. Mechanism D: same as Mechanism C, the arithmetic crossover is used with a modified cost function based on the number of genes that are hidden. The offspring is biased toward better parent (lower cost) with less hidden genes.

Mechanism A is schematically shown in **Figure** 6.

As shown in **Figure** 6, the tags can do crossover between themselves regardless the genes. In the figure, the tag strings are broken in two points (tag3-tag4 and tag5-tag6), while the gene strings are broken in one point (gene2-gene3). The crossover points for genes or tags are chosen stochastically.
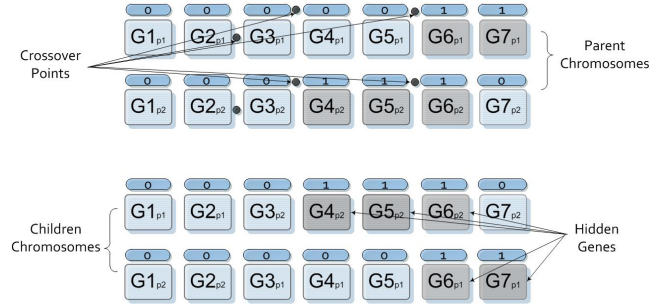
Figure 6: Schematic of Mechanism A

Figure 7 shows the intermediate offsprings in Mechanism B. After these chromosomes are built, the cost of each chromosome is calculated. The final offspring (output of crossover function) is the arithmetic weighted average of parents, with more weight towards the parent with less cost for its intermediate offspring. For example, if in **Figure** 7, intermediate offspring 1 has less cost, then the final offspring would be the arithmetic weighted average of parents, closer to parent 1.

Figure 7: Schematic of Mechanism B

The cost functions of Mechanism C and D are as follows:
Mechanism C:

$$f_{modified}(X) = f(X) - \sum_{i=1}^{M}(flag_i). \qquad (2)$$

Mechanism D:

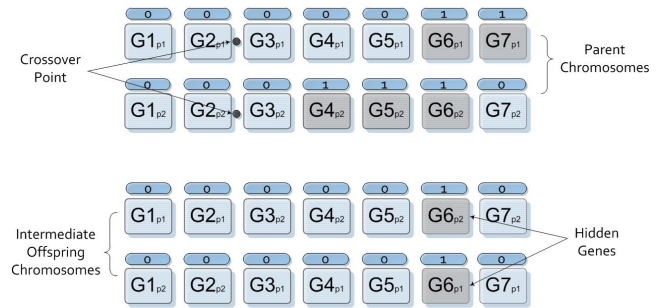$$f_{modified}(X) = f(X) + \sum_{i=1}^{M}(flag_i). \qquad (3)$$

where M is the number of genes that can be hidden. Comparison between the four mechanisms is conducted through testing (as detailed in Section 3). The results show that Mechanism A performs better than other mechanisms in general. The detailed results and simulations are presented in next sections.

3. TEST CASES: 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, the Styblinski-Tang, and the Ackley 4 functions. The general concept of modifying these functions to be VSDS functions is here described. Consider the optimization cost function defined as:

$$F(X) = \sum_{i=1}^{N} f_i \qquad (4)$$

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 4. Unlike the hidden gene tags, the chromosomes evolve through the standard GA selection, mutation and crossover operations. For the chromosomes, a single point crossover and an adaptive feasible mutation operators are selected. The GA parameters used in these simulations are listed in Table 2.

Table 1: Genetic Algorithm Options in Matlab

Option	Value
$Population\ Size$	400
$Number\ of\ Generation$	50
$Mutation\ Probability$	0.01
$Elite\ Count$	20
$Crossover\ Fraction$	0.95
$TolFun$	$1e-6$
$TolCon$	$1e-6$

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 $f(x_j^*) = 0$, $\forall j \in \Gamma$.

3.1 Results Using Stochastically Evolving Tags

All the four mechanisms are tested on the four selected mathematical optimization functions. Each test case is simulated 20 times, with population size of 400 and 50 number

of generations. The function equations and upper and lower boundaries on variables can be found in [11]. In Equation 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. Based on the simulation results (first four rows in Table 6), Mechanism A is the best choice among four proposed mechanisms for 2 out of 4 test cases. Moreover, comparing the four proposed mechanisms, Mechanisms C and D result in highest cost function with lowest occurrence probability in all the cases. A more detailed analysis of the results is conducted in Section 5.

4. TEST CASES: INTERPLANETARY TRAJECTORY OPTIMIZATION PROBLEMS

The optimization of a space trajectory is a complex problem that requires topology optimization. Specifically, to define a trajectory for a space mission from Earth to Jupiter, it is required to determine the topology of the solution which includes the sequence of fly-by planets and the number of Deep Space Maneuvers (DSMs). In addition to the topology optimization, it is required to determine the time of fly-bys, the hight and inclination of these fly-bys among other variables that are listed in Table 3 [6]. This problem is used to test the ability of the HGGA, with the new definitions of hidden genes tags and evolution mechanisms, to autonomously search for the optimal topology.

4.1 Earth - Jupiter Mission Trajectory Optimization

The Earth-Jupiter mission is optimized with the concept of HGGA with tags and the results are compared to the results in literature. Genetic Algorithm options in Matlab are also presented in Table 2 and the upper and lower ranges of design variables are listed in Table 3. A detailed expression of the problem and the its orbital mechanics can be found in [6].

Table 2: Genetic Algorithm Options in Matlab

Option	Value
$Population\ Size$	400
$Number\ of\ Generation$	50
$Mutation\ Probability$	0.01
$Elite\ Count$	20
$Crossover\ Fraction$	0.95
$TolFun$	$1e-6$
$TolCon$	$1e-6$

The problem is solved in two steps. First, a zero Deep Space Maneuver (DSM) trajectory is considered where it is assumed that there is no DSM in the trajectory and the algorithm searches for the optimal trajectory with no DSMs. This zero-DSM solution is then used to solve the second step - the Multi Gravity Assist with Deep Space Maneuvers (MGADSM) mission where DSMs are included and the all variables are optimized holding the fly-by planets sequence fixed. This two phase approach is detailed in [6], and has shown to be computationally efficient. The number of generations and population size are selected to be 100 and 300 for zero-DSM and 100 and 500 for MGADSM phases, respectively, and each phase is simulated for 100 times.

Table 3: Lower and upper bounds of Earth-Jupiter problem

Design Variable	Lower Bound	Upper Bound
Number of flybys	0	2
Flyby 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
Flyby normalized pericenter altitude	0.1	10
Flyby normalized pericenter altitude	0.1	10
Flyby plane rotation angle (rad)	0	2π
Epoch of DSM	0.1	0.9
DSM (km/s)	-5	5

Figure 9: Mechanism A: EVEJ Trajectory for MGADSM Model

4.1.1 Mechanism A: Earth - Jupiter Mission Trajectory Optimization

The two phases are solved using Mechanism A for hidden gene assignment, and the results are presented in Table 4.

The mission trajectories for zero-DSM and MGADSM models are shown in **Figure** 8 and **Figure** 9.

the proposed hidden genes assignment mechanisms can autonomously find the solution topology.
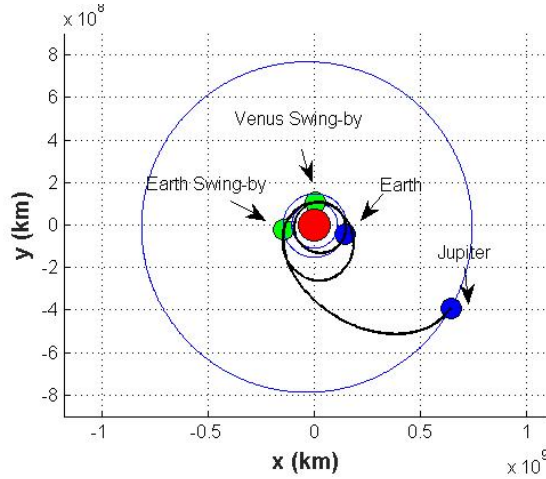

Figure 8: Mechanism A: EVEJ Trajectory for Zero-DSM Model

As shown, flyby scenario is found similar to [6], which used a feasibility criterion for HGGA assignment. Moreover, the cost of zero-DSM phase is 10.1607 (km/s), and the cost of MGADSM phase is 10.1438 (km/s). Both phases have lower cost compared to the values reported in [6]. This shows the ability of this mechanism to find suitable solutions.

4.1.2 Mechanism B: Earth - Jupiter Mission Trajectory Optimization

First, the zero-DSM problem is solved and the results of this model are used for the MGADSM model. The results of both phases are presented in Table 5.

The mission trajectories for zero-DSM and MGADSM models are shown in **Figure** 10 and **Figure** 11. The total cost of the mission in the solution obtained using Mechanism B is 10.9822 (km/s). These results demonstrate that
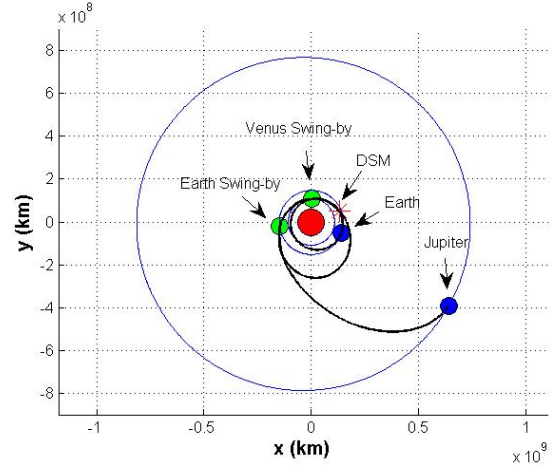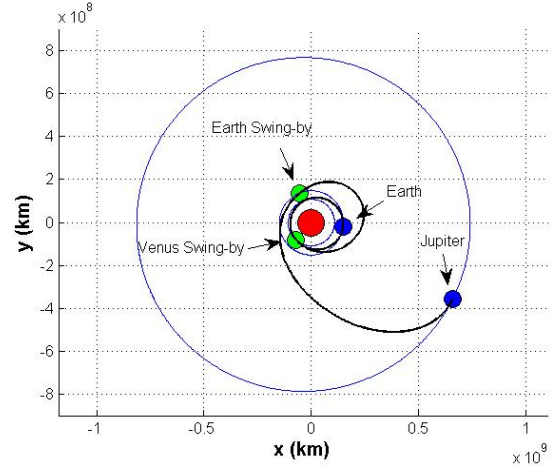

Figure 10: Logic B: EVEJ Trajectory for Zero-DSM Model

The results of MGADSM problem is consistent with the result of mathematical function. In both problems, Mechanism A performs better than Mechanism B. Mechanisms C and D are also tested and the results show that although they can find the Earth-Venus-Earth-Jupiter flyby sequence, they are unable to find low cost trajectories compared to Mechanisms A and B. Mechanism C results in 15.8764 km/sec for zero-DSM model and 12.3237 km/sec for MGADSM model. Also Mechanism D results in 15.9100 km/sec for zero-DSM model and 14.9624 km/sec for MGADSM model.

5. STATISTICAL ANALYSIS

The results of the experiments need to be analysed regarding the performance of the suggested mechanisms. First analysis is on their success rate. For each mathematical test case, the success rate of each mechanism is calculated numerically. **Figure** 12 shows the success rate of Styblinski-

Table 4: HGGA solution of Earth-Jupiter problem using Mechanism A

Mission parameter	Zero-DSM model (first step)	MGADSM model (second step)
Departure Date	$04 - Sep - 2016, 14 : 35 : 18$	$02 - Sep - 2016, 04 : 38 : 48$
Departure Impulse (km/s)	3.5029	3.3084
DSM date	–	$16 - Oct - 2016, 12 : 13 : 23$
DSM impulse (km/s)	–	0.17826
Venus flyby date	$05 - Sep - 2017, 09 : 07 : 43$	$06 - Sep - 2017, 15 : 18 : 56$
Post-flyby impulse (km/s)	$5.5107e - 4$	-0.006971
Pericenter altitude (km)	1257.3126	821.6738
DSM date	–	–
DSM impulse (km/s)	–	–
Earth flyby date	$29 - Mar - 2019, 05 : 21 : 55$	$28 - Mar - 2019, 15 : 28 : 42$
Post-flyby impulse (km/s)	0.4424	0.4431
Pericenter altitude (km)	637.8000	637.8000
Arrival date	$03 - Sep - 2021, 15 : 37 : 27$	$05 - Sep - 2021, 04 : 14 : 00$
Arrival impulse (km/s)	6.2129	6.2022
TOF (days)	$365.7725, 569.8432, 889.4275$	$369.4445, 568.0068, 891.5315$
Mission duration (days)	1825.0432	1828.9828
Mission cost (km/s)	10.1607	10.1438

Table 5: HGGA solution of Earth-Jupiter problem using Mechanism B

Mission parameter	Zero-DSM model (first step)	MGADSM model (second step)
Departure Date	$14 - Sep - 2016, 01 : 15 : 26$	$12 - Sep - 2016, 02 : 49 : 51$
Departure Impulse (km/s)	3.3086	2.5675
DSM date	–	$11 - May - 2017, 12 : 17 : 00$
DSM impulse (km/s)	–	1.4871
Venus flyby date	$01 - Dec - 2017, 16 : 40 : 18$	$28 - Nov - 2017, 11 : 19 : 02$
Post-flyby impulse (km/s)	1.3693	-0.001062
Pericenter altitude (km)	13968.4261	2969.6685
DSM date	–	–
DSM impulse (km/s)	–	–
Earth flyby date	$10 - Jan - 2019, 23 : 39 : 16$	$09 - Jan - 2019, 06 : 28 : 54$
Post-flyby impulse (km/s)	0.6814	0.6588
Pericenter altitude (km)	637.8000	637.8000
Arrival date	$07 - Oct - 2021, 12 : 00 : 48$	$08 - Oct - 2021, 00 : 32 : 10$
Arrival impulse (km/s)	6.2142	6.2423
TOF (days)	$443.6428, 405.291, 1000.515$	$442.3536, 406.7985, 1002.7523$
Mission duration (days)	1849.4482	1851.9044
Mission cost (km/s)	12.07771	10.9822
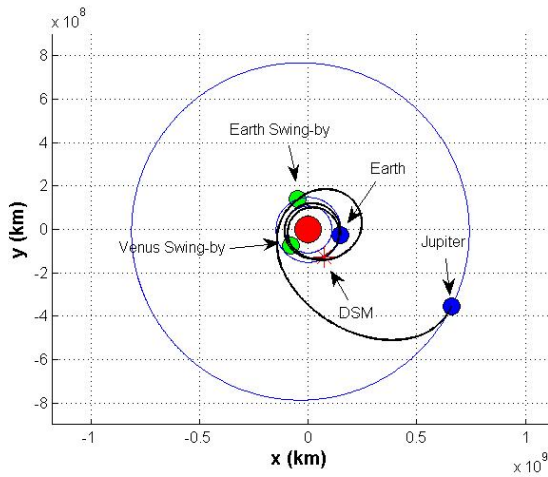
Figure 11: Logic B: EVEJ Trajectory for MGADSM Model

Tang function as an example. In all four cases, Mechanism A has the highest success rate but there is no obvious success rate ranking among Mechanism B, Mechanism C, and Mechanism D. Moreover, almost in all cases, the difference between the success rate of Mechanism A and other three mechanisms is significant, while Mechanism B, Mechanism C, and Mechanism D converge closely.

For further analysis, two ranking method are selected for static analysis: the Sign test, and the Friedman rank test [9]. The Sign test is a pairwise comparison, while the Friedman rank test is a multiple comparison where the joint analysis of methods is carried out. The complete description of these tests can be found in [9, 5].

In the Sign test, each mechanism is ranked based on the total number of cases in which the mechanism works the best (results in lower cost value). Table 6 shows the comparison of four HGGA mechanisms and standard HGGA for all the mathematical functions and the interplanetary trajectory optimization problem (Zero-DSM and MGADSM models).

In Table 7, the Sign rank of each method is presented.

For a mechanism to be significantly better, for n test cases, there should be at least $n/2 + 1.96\sqrt{n}/2$ cases where the

Table 6: Cost values of all the test cases for different mechanisms

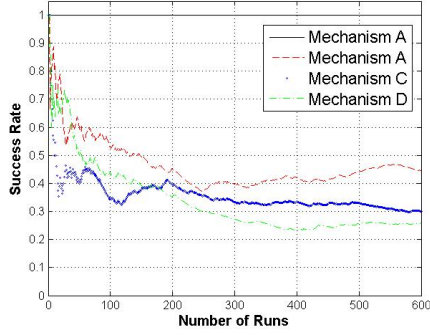|	Mechanism A	Mechanism B	Mechanism C	Mechanism D	Standard HGGA
Egg Holder function	$-3552.8947, 50\%$	$-3644.2279, 50\%$	$-2571.8028, 10\%$	$-2134.7217, 10\%$	$-2749.2646, 30\%$
Schwefel 2.26 function	$-418.2850, 100\%$	$-418.0799, 100\%$	$-375.3514, 5\%$	$-303.5320, 10\%$	$-335.1844, 100\%$
Styblinski-Tang function	$-195.8298, 100\%$	$-195.8278, 100\%$	$-191.8393, 10\%$	$-194.5325, 10\%$	$-156.6646, 100\%$
Ackley 4 function	$-12.2281, 10\%$	$-12.4931, 20\%$	$-11.6045, 10\%$	$-11.8215, 10\%$	$-7.8898, 95\%$
Earth-Jupiter problem (zero-DSM)	10.1607	12.0777	15.8764	15.9100	10.298
Earth-Jupiter problem (MGADSM)	10.1438	10.9822	12.3237	14.9624	10.182

Figure 12: Success rate vs. number of runs for Styblinski-Tang function in different mechanisms

Table 7: Sign test ranks

	Number of wins
Mechanism A	4
Mechanism B	2
Mechanism C	0
Mechanism D	0
Standard HGGA	0

mechanism results in lowest cost value. Here, for $n = 6$, a mechanism should win all the test cases to be significantly better than other mechanisms. Mechanism A wins in 4 cases, so it is not significantly better, although it has the highest rank among all the mechanisms.

In Friedman rank test, the methods are ranked first, and then the Friedman statistic F_f will be calculated. The null hypothesis in Friedman rank test is that there is no significant difference between the methods. If the p-value (calculated probability) is lower than the significance level, the null hypothesis is rejected, meaning that there is a significant difference between the methods. The Friedman statistic is calculated as follows [5]:

$$F_f = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1). \quad (5)$$

where R_j is the summation of rank of method j, k is the number of methods, and n is the number of test cases. The distribution of F_f is according to χ^2 distribution with $k - 1$ degrees of freedom [5], when k and n are big enough. Therefore, for the significance level of α, if the computed F_f value is greater than the upper-tail critical value for the χ^2 distribution, the null hypothesis is rejected. This concept is shown in **Figure** 13.
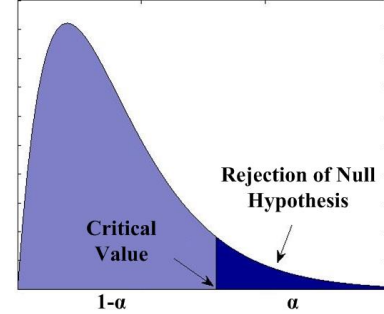

Figure 13: Rejection of null hypothesis in Friedman test based on critical value of χ^2 distribution

Table 8 shows the ranks of the methods for each test case and also the summation of ranks. As shown, Mechanism A has the best average rank. By substituting in Equation 5, the Friedman statistic will be $F_f = 15.6$ and its p-value will be 0.0036. Since the p-value is less than significance level of $\alpha = 0.05$, and F_f is greater that the critical value of χ^2 distribution (which is 9.48773 for 4 degree of freedom and $\alpha = 0.05$), it is concluded that the null hypothesis is rejected and there is significant difference between the methods.

6. CONCLUSIONS

The hidden genes genetic algorithms proposed recently for systems architecture optimization is studied in this paper. Specifically the mechanisms for assigning the hidden genes in a chromosome are investigated. This paper introduces the concept of binary tags that assign genes to be either hidden or active. 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 another cell. Four mechanisms for evolving the values of these tags are proposed in this paper that use a stochastic approach for tags evolution. Numerical tests were conducted on mathematical optimization problems as well as the interplanetary trajectory optimization mission to planet Jupiter.

The implementation of the new hidden genes assignment mechanisms to the space trajectory optimization problem and the mathematical optimization problems demonstrated its capability in finding the optimal number of design variables. For the trajectory optimization problem, it is possible to autonomously compute the optimal number of swing-bys, the planets to swing-by, and the optimal number of deep space maneuvers, in addition to the rest of the design variables using the proposed hidden genes concept. One of the mechanisms resulted in improving the solution (finding lower

Table 8: Friedman ranks

	Mechanism A	Mechanism B	Mechanism C	Mechanism D	Standard HGGA
Egg Holder function	2	1	4	5	3
Schwefel 2.26 function	1	2	3	5	4
Styblinski-Tang function	1	2	4	3	5
Ackley 4 function	1	2	4	3	5
Earth-Jupiter problem (zero-DSM)	1	3	4	5	2
Earth-Jupiter problem (MGADSM)	1	3	4	5	2
R_j	8	12	23	26	21

cost) compared to the original hidden genes approach that does not implement the tags mechanisms.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] O. Abdelkhalik. Hidden genes genetic optimization for variable-size design space problems. *Journal of Optimization Theory and Applications*, 156(2):450–468, 2013.

[2] J. T. Allison, A. Khetan, and D. Lohan. Managing variable-dimension structural optimization problems using generative algorithms. In 10th *World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, May 19 - 24 2013.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[4] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26:29–41, 1996.

[5] M. FRIEDMAN. The use of ranks to a void the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200), 1937.

[6] A. Gad and O. Abdelkhalik. Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization. *AIAA Journal of Spacecraft and Rockets*, 48(4):629–641, JulyŰ-August 2011.

[7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[8] X. Guo, W. Zhang, and W. Zhong. Topology optimization based on moving deformable components: A new computational framework. *Computing Research Repository*, abs/1404.4820, 2014.

[9] D. M. F. H. Joaquin Derrac, Salvador Garcia. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Journal of Swarm and Evolutionary Computation*, 1, 2011.

[10] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceeding of the 1995 IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995, 1995. IEEE Service Center.

[11] X.-S. Y. Momin Jamil. A literature survey of benchamrk functions for global optimization problems. *Int. Journal of Mathematical Modeling and Numerical Optimizatin*, 4(2), 2013.

[12] K. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Natural Computing Series. Springer, 2005.

[13] B. Starr. Spooled dna and hidden genes: The latest finding in how our dna is organized and read. *The Tech Museum of Innovation, Department of Genetics, Stanford School of Medicine, 201 South Market Street San Jose, CA 95113*. http://www.thetech.org/genetics/news.php?id=31.

[14] N. I. o. H. The National Institute of General Medical Sciences. The new genetics. On line website, April 2010.

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