Design optimization of actuator layouts of adaptive optics using a genetic algorithm

Gregory J. Michels*, Victor L. Genberg, Keith B. Doyle, Gary R. Bisson
Sigmadyne, Inc.
803 West Avenue, Rochester, NY 14611

ABSTRACT

The development of the optimum locations of actuators for an adaptive optic has in the past been a manually iterative process. A more automated yet efficient method is desired to quickly develop actuator layouts and the associated optical performance. A genetic design optimization algorithm is developed and implemented in software. The method is then demonstrated on an adaptive optic to show how it can be used to develop optimum actuator layouts for a fixed number of actuators or to conduct design trades in choosing the number of actuators.

Keywords: Finite Element, Optimization, Opto-mechanics, Adaptive, Deformable

1. INTRODUCTION

The development of the optimum locations of actuators for an adaptive optic has in the past been a manually iterative process. A more automated yet efficient method is desired to quickly develop actuator layouts and the associated optical performance. Such a method must be able to consider multiple disturbances cases for which correction is desired and must allow control over the number of actuators allowed. A genetic design optimization algorithm is developed and implemented in *SigFit*[1] since the capabilities of this software already include simulation of adaptive control. The method is then demonstrated on an adaptive optic to show how it can be used to develop optimum actuator layouts for a fixed number of actuators or to conduct design trades in choosing the number of actuators.

2. OVERVIEW OF GENETIC ALGORITHMS

2.1 What are genetic algorithms?

Genetic algorithms are very robust optimization methods.[2] Their design, if implemented correctly, enables them to find global optimums even in design spaces containing many local optimums. The basic method can be broken up into the following operations: (a) initial population generation, (b) fitness evaluation, (c) convergence evaluation, (d) mating selection, and (e) crossover and mutation. The population at any time during the process is a finite set of designs currently under consideration. Each design in the population, called an *individual*, is defined by a series of binary digits called a *chromosome*. The chromosome is simply a string of 1's and 0's which are interpreted to be a description of a physical design in some fashion. The sequence of five basic phases listed above are performed repeatedly to develop new populations called *generations*. The process ends when the convergence evaluation determines that production of more generations is no longer beneficial. The optimum is the individual with best performance, or *fitness*, over all individuals considered in the process.

2.2 Why a genetic algorithm?

Before developing a genetic optimization method for finding optimum actuator layouts, one might be tempted to suggest that it would be easier to use a full permutation analysis in which all possible arrangements of actuators are considered. For the purposes of testing the genetic algorithm such a routine was written to find the true global optimum and to understand the entire design space by a full permutation method. With this information about the entire design space we are able to assess the effectiveness of the genetic algorithm. However, while it is tolerable to use this method on test cases used to validate the effectiveness of the genetic algorithm, use of this method for actual problems is not desirable since the genetic algorithm is

* Correspondence: Email: michels@sigmadyne.com; www: http://www.sigmadyne.com; Telephone: (585) 235-6892

consistently able to find the global optimum in a small fraction of the time (seconds) that the full permutation run takes (hours or even days for large problems).

3. GENETIC ALGORITHM FOR ACTUATOR PLACEMENT

The implementation of a genetic algorithm to the application of adaptive optic actuator selection requires special considerations but the basic method is the same as that found in the referenced literature. A finite element model of the optic to be analyzed is constructed for the purpose of computing actuator influence function and operational disturbance predictions. A set of candidate actuator locations is selected at locations where actuators could be placed. The finite element model is then exercised to compute the optical surface nodal displacements for all of the candidate actuator influence functions and the operational disturbances.

3.1 Definitions

As actuator layouts are developed within the algorithm, the selections of actuators in each layout are made such that the desired symmetry in the actuator layout is maintained. In addition, a user specifiable number of actuators is preserved, and a set of "always present" actuators is always included. The symmetry in each actuator layout is maintained by allowing the user to define links between the inclusion of *slave actuators* with the inclusion of *master actuators*. Figure 1 shows an example of an adaptive optic with candidate actuator locations. The master actuators are shown in solid black while the slave actuators are shown in gray. Some of the master and slave actuator links are shown with alphabetic labels. For example, if master actuator "A" is included in a particular actuator layout then the five other slave actuators labeled "A" are included in the layout as well. The set of a master actuator and its associated slave actuator is called a *variable group*. With the requirement that all groups must contain the same number of actuators, the number of actuators is maintained by including only a user specified number of groups in each actuator layout. Separately the user may specify a set of actuators which must always be included. An example of such a set of actuators would be three displacement actuators to mount a force actuated mirror. This set of actuators is called the *fixed group*.

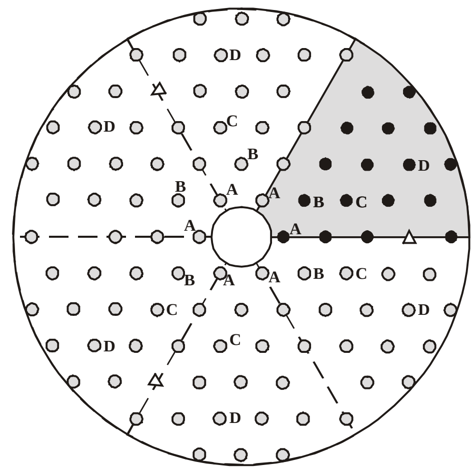

Figure 1: Example adaptive optic with candidate actuator locations. The master force actuator candidate locations are shown with black circles, the slave force actuator candidate locations are shown with gray circles, and the displacement actuators are shown with white triangles.

3.2 Initial population generation

An initial population of an even number of individuals is randomly selected. Measures are taken to make sure that all individuals have only the number of variable groups specified by the user. The included variable groups are recorded for each actuator by a chromosome which has a digit length equal to the number of candidate variable groups. With each digit associated with a particular variable group, a "1" indicates the variable group is included in the individual while a "0" indicates that it is not. The number of individuals in the initial population is a user specifiable quantity and can affect the success of the optimization process.

3.3 Fitness evaluation

Once the initial population is generated a measure of performance is computed for each individual. This measure is called the *fitness*. For adaptively corrected optics the fitness is chosen to be the corrected surface error. The adaptively corrected surface error is computed for each individual by methods available in the literature.[3] Before each individual's fitness is computed, a check is made to determine if we have already evaluated the fitness of the same individual. If it has already been computed, it is simply recalled from memory instead of performing a duplicate adaptive control simulation. This is done to increase the efficiency of the process.

3.4 Convergence evaluation

Once the fitnesses of the current generation have been found, they may be compared to prior generations to evaluate convergence. There is considerable flexibility in how this task is executed. For our algorithm the user is able to specify relative and absolute convergence criteria to be met over a user specifiable number of generations. The absolute and relative changes in the best known fitnesses are recorded between successive generations. If either of these changes are less than the user specified values, the process is stopped and the best known fitness and associated individual is reported.

3.5 Mating selection, crossover, and mutation

If convergence has not been achieved then members of the current generation are selected for mating. The number of members of the current generation which are selected for mating is the same as the number of individuals in the population. The selections are made by a random process weighted by the relative fitnesses of all of the individuals. The weighting factor used for each individual is given in Eq. (1),

$$W_i = \frac{1/X_i}{\sum_{j=1}^{N} 1/X_j} \tag{1}$$

where, W_i is the relative weighting factor of the ith individual, X_i is the corrected surface RMS error of the ith individual, and N is the number of individuals in the current generation. The members of the current generation which are chosen for mating are stored in a mating pool.

Once the mating pool is generated, crossover is performed. To perform crossover members of the mating pool are taken in sequence two at a time. The chromosomes of each pair are scanned and compared gene by gene. Unlike genes are exchanged between the chromosomes with some user specifiable probability. This crossover is called *primary crossover*. If a gene is found to be the same between the two mates, the gene is skipped and the next gene of the two mates are examined. Notice that the primary crossover operation adds an actuator group to one individual while subtracting it from another. Therefore, a *companion crossover* operation must be performed on any of the gene sites which will rebalance the number of variable groups in each individual. This is done by searching for all genes sites, excepting the primary crossover site, which, if exchanged, will rebalance the number of variable groups in the individuals to the correct value. One of these sites is chosen at random and companion crossover is performed. If no genes sites are found which will rebalance the number of variable groups the primary crossover operation is reversed. The process continues with examining the next gene for primary crossover until the end of the chromosomes are reached. The resulting two chromosomes are two individuals in the next generation.

This process is performed for the entire gene pool two individuals at a time until a new generation has been created. It is in this operation that the requirement of a even number of individuals in a population originates.

Once the new generation is complete a mutation operation is performed. Each gene in each individual is examined and is reversed with a small probability. If a gene is successfully chosen for mutation it is called *primary mutation*. Then as was done in the crossover operation, a companion mutation operation is performed to balance the number of included variable groups. Once mutation is complete, the cycle is restarted in fitness evaluation with the new generation.

4. EXAMPLE

To illustrate the effectiveness of the genetic algorithm a simple example was developed.

4.1 Problem definition

The optic for which an optimum actuator layout is to be determined is shown in Figure 2 with dimensions. The optic is a lightweighted mirror comprised of a triangular-cell core sandwiched by two faceplates. The mirror is supported on three stiff displacement actuators. A set of 210 candidate low stiffness force actuator locations were selected to be at strut intersections and are shown in Figure 3. These actuators are intended to be push-pull style force actuators acting along the optical axis. In order to maintain symmetry, 35 master candidate locations were chosen at the locations shown with black circles while the remaining locations were defined as slave candidate locations. This created 35 variable groups of six locations each. It was desired to find the optimum layout which uses a total of 18 actuators. Therefore, only three variable groups were allowed in any actuator layout considered by the genetic algorithm. The three displacement actuators shown with black triangles were defined as a fixed group present in all layouts.

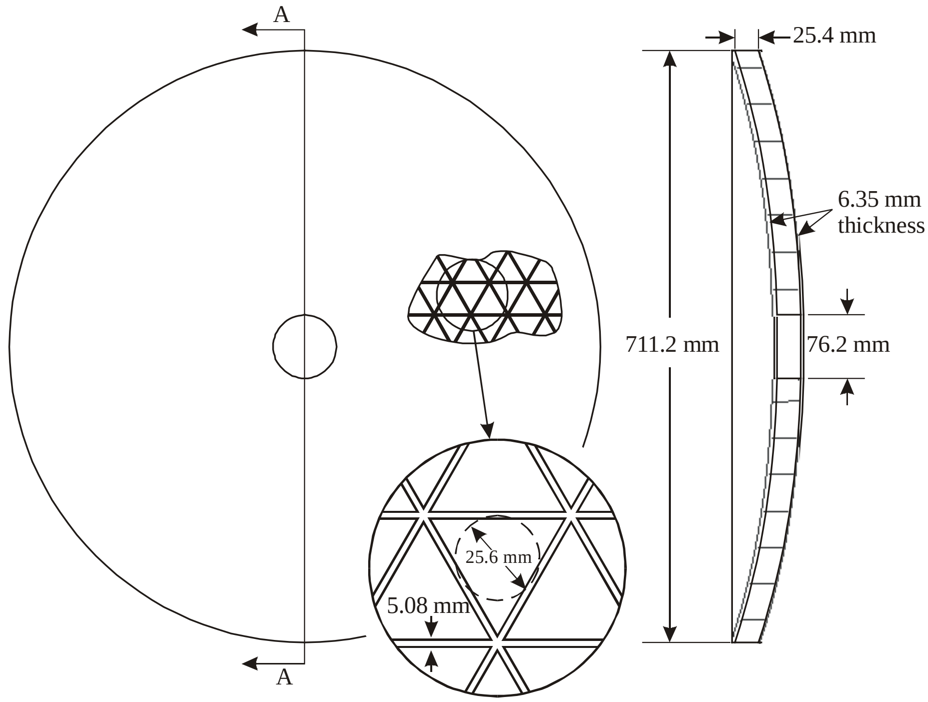

Figure 2: Drawing of test model lightweighted mirror.

Three independent disturbance conditions were considered separately and then combined as a fourth case to demonstrate the flexibility and power of the genetic algorithm in finding optimums. The first disturbance condition is a gravity load associated with moving the mirror to a zero gravity operating environment as would be expected for an orbiting telescope. The second disturbance is an isothermal drop in temperature of 20°C while the third disturbance is a front-to-back temperature difference of 0.15 °C.

4.2 Benchmarking the problem by full permutation

In the evaluation of an optimization method's effectiveness, it is advantageous to compare the results of the optimization algorithm with knowledge of the entire design space. In many problems this is impossible or prohibitive because the feasible design space contains either infinite designs or a number of designs that makes complete evaluation of the design space prohibitive. Yet, with 35 variable groups it is a tolerable task to compute the corrected surface error of every actuator layout of 18 actuators for each of the four load case conditions. The genetic algorithm subroutine in *SigFit* was replaced by a subroutine which finds and evaluates every actuator layout. For this 35 candidate variable group problem with three allowable variable groups, the full-permutation analysis requires consideration of 6545 layouts and can be run in 28 minutes on a Pentium 4 3.2 GHz processor with 1 GB of RAM.

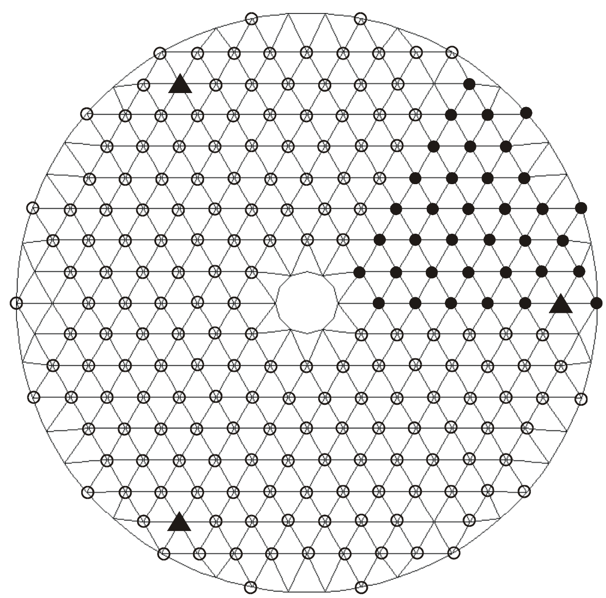
Figure 3: Candidate actuator layout.

The results of the full permutation analyses are shown in the four histogram plots in Figure 4. Each bar represents the number of actuator layouts having a corrected surface RMS error within the 5 nm band below the label on the horizontal axis. For example, a bar above the label 50 shows the number of layouts having corrected surface RMS errors between 45 nm and 50 nm. The goal of the genetic optimizer is to find the actuator layouts within the left-most bar of the histogram. Since there are relatively few designs in the left-most bar of each histogram as compared to the layouts in the full-permutation, this will be a discerning test of the genetic optimizer.

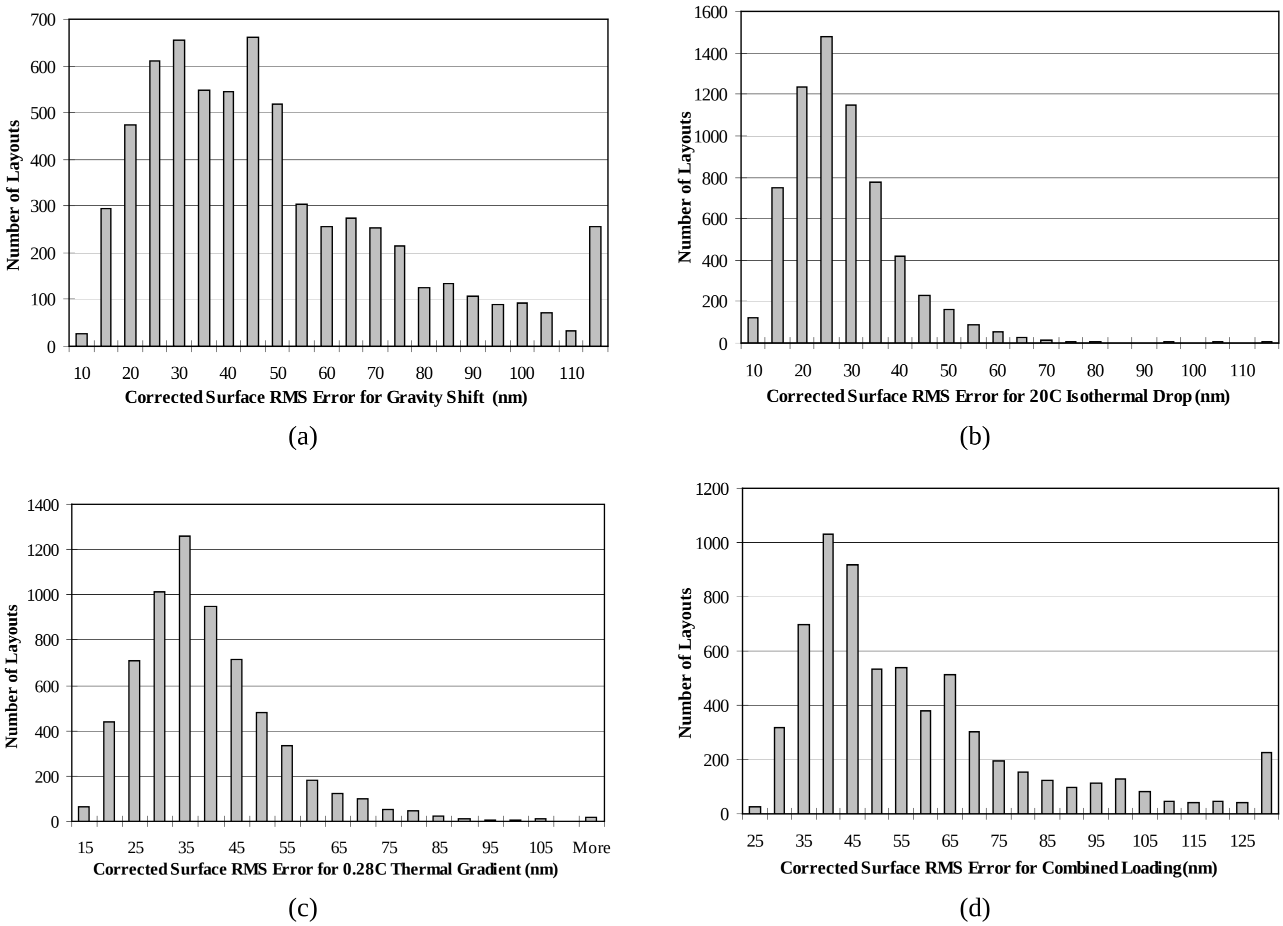

Figure 4: Histogram plots of all 6545 designs with 18 actuators showing number of actuator layouts vs. corrected surface RMS error in nanometers. Each bar represents a 5 nm surface RMS error bandwidth. The last bar represents the number of designs with corrected surface RMS errors above the last number shown on the horizontal axis.

4.3 Results from the genetic algorithm

The genetic algorithm was executed four times for each load case with the parameters shown in Table 1.

Table 1: Genetic Algorithm Parameter Values

Parameter	Value
Population Size	30
Probability of Crossover	0.6
Probability of Mutation	0.0333
Maximum Relative Convergence	0.01
Number of Generations Maximum Relative Convergence Must be Met	4
Maximum Absolute Convergence	0.1
Number of Generations Maximum Absolute Convergence Must be Met	4

Table 2 shows the results of four genetic optimization executions compared to the global optimum and global average found from the full-permutation analysis for the 35 variable group problem defined in Section 4.1. Notice that since the initial population generation and subsequent crossover and mutation operations are random processes, the result of each optimization varies from run to run. However, in comparison to the histogram plots in Figure 4, the resulting optimum actuator layouts are hardly random results. The optimums are consistently very close to the global optimum. In fact, the results from all analyses are consistently superior to 99% of all possible actuator layouts, and the best of the four executions for each load condition are all within the left-most bar of the corresponding histogram in Figure 4.

Table 2: Corrected Surface RMS Errors of Layouts found by Genetic Optimization Compared to Global Results

Operational Disturbance	Genetic Result #1	Genetic Result #2	Genetic Result #3	Genetic Result #4	Best Analysis Result	Global Optimum	Global Average
Gravity Variation	8.4 nm	9.1 nm	10.1 nm	8.5 nm	8.4 nm	7.0 nm	47.8 nm
20 °C Isothermal Temperature Drop	10.8 nm	9.5 nm	10.6 nm	12.2 nm	9.5 nm	6.1 nm	25.9 nm
0.15 °C Front-to-Back Axial Temperature Difference	17.7 nm	18.1 nm	13.9 nm	15.6 nm	13.9 nm	10.6 nm	36.6 nm
Combination of Above	27.2 nm	26.6 nm	24.2 nm	25.4 nm	24.2 nm	21.6 nm	57.3 nm

To improve on the performance of the genetic algorithm to find a truly global optimum, the process parameters can be adjusted. For example, the population size may be increased or the number of generations over which the convergence conditions must be met may be increased.

The actuator layouts for the optimum found by the genetic algorithm for each load case are shown in Figure 5. Each optimum actuator layout is arranged to correct for its given load condition. The layout in Figure 5(a) best corrects for the gravity shift by a layout which can correct a large amount of trefoil deformation. For the axial thermal gradient case the optimum actuator layout is best able to generate edge moments which one would expect from the theory of elasticity. The arrangement for the combined loading is a mix of the arrangements for the individual loading conditions.

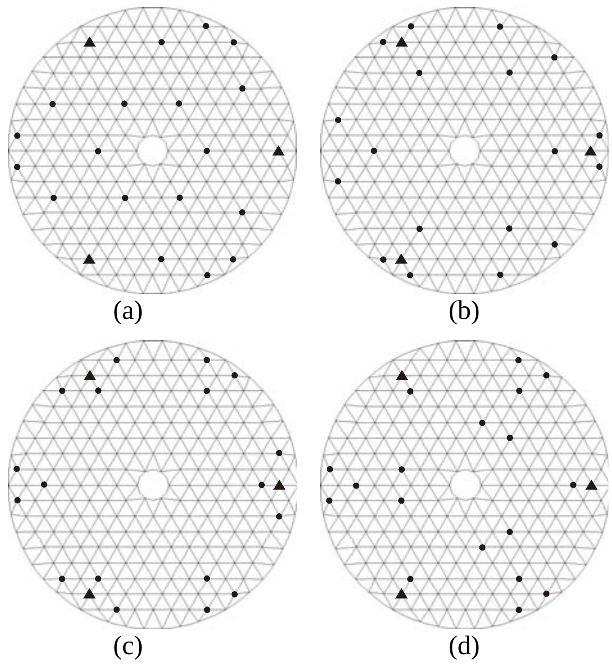

Figure 5: Optimum actuator layouts found by the genetic algorithm for deformations generated by (a) gravity shift, (b) isothermal shift, (c) axial gradient, and (d) combined loading.

4.4 Number of actuators trade study

If it is desired to understand the minimum number of actuators to use in an adaptive optic application, the genetic algorithm can be used to develop a useful design curve for this purpose. The genetic algorithm was used to find optimum actuator layouts with various numbers of allowed variable groups for the combined loading condition. The resulting design curve shown in Figure 6. A trade study of this type is very useful to find minimum number of actuators required to achieve a desired corrected surface performance. For example, if the requirement for this mirror is to correct to within 40 nanometers of surface RMS, then an arrangement using only 12 actuators would reduce the weight compared to an 18 actuator design.

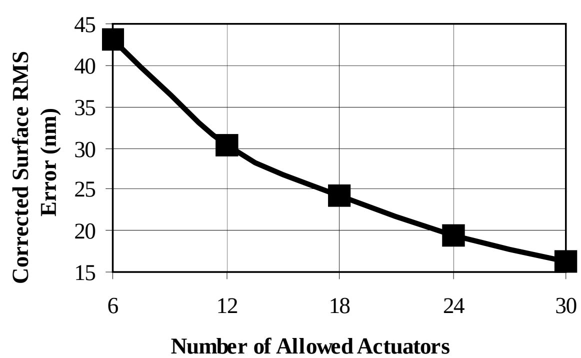

Figure 6: Corrected surface error vs. allowed actuators.

4.5 Solving larger problems

As was mentioned above, the task of running full permutation optimization in lieu of a more sophisticated algorithm becomes increasingly prohibitive and the usefulness of the genetic algorithm becomes more clear. For the particular application at hand Equation 2 gives the expression for the total number of possible actuator layouts

$$N_T = \frac{N_v!}{N_a!(N_v - N_a)!} \tag{2}$$

where, N_T is the total number of actuator layouts with N_a allowed groups and N_v is the number of candidate variable groups. With N_v equal to 35 and N_a equal to 3, there are 6545 allowable actuator layouts out of the 1 to 2^{35} = 34,359,738,368 number line representable by the 35 digit long chromosome. The number of allowable actuator layouts increases by an order of magnitude for every additional allowable group as shown in Figure 7(a) while the number of chromosomes over which the allowable actuator layouts must be found increases by a factor of two as shown by the dashed curve in Figure 7(b). Both of these quantities contribute to dramatic increases in computational expense in evaluating the full permutation of the design space for this problem.

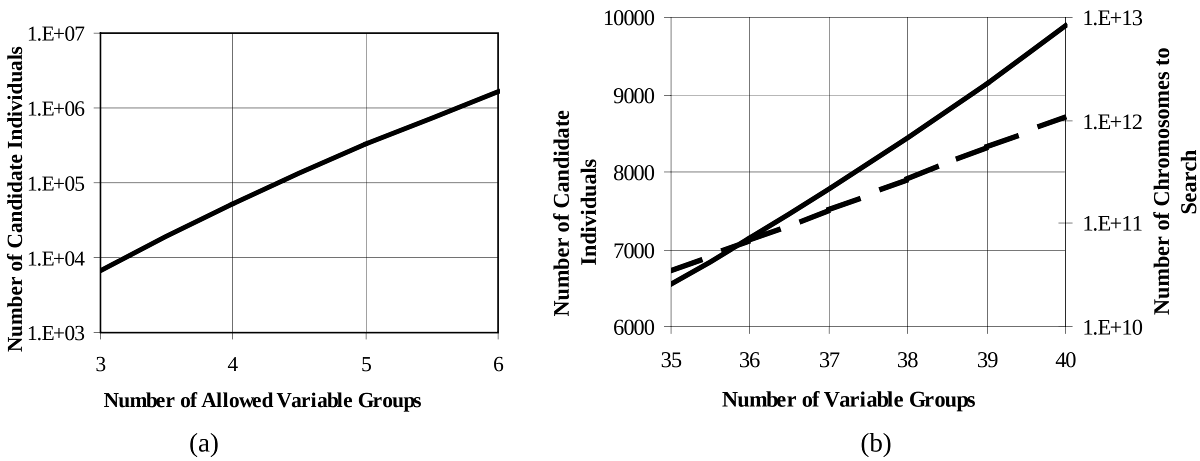

Figure 7: Effects of number of allowed groups and number of variable groups on full permutation problem size. (a) Number of candidate individuals vs. number of allowed variable groups with 35 candidate variable groups. (b) Number of candidate individuals (solid curve) and number of chromosomes to search (dashed curve) vs. number of variable groups with 3 allowed variable groups.

The test case described above has been sized such that it is large enough to be meaningful, yet, small enough that knowledge of the entire space is attainable. But as the candidate variable group size is increased this full-permutation evaluation time increases quickly. Each added variable group doubles the analysis time. Therefore, consideration of candidate actuator locations at all core strut intersections of the example mirror requires 40 groups and about 15 hours of full-permutation analysis time. If the candidate actuator locations are not in discrete spots but instead continuously variable, then the number of variable groups desired may be much higher in order to obtain more precise locations of the actuators in an optimum layout. Yet, for this considerably larger optimization problem the genetic optimizer was found to produce a repeatable optimum in a few seconds per analysis. It is in these full-permutation tests compared to the genetic analysis tests that the efficiency of the genetic optimizer is fully demonstrated.

5. SUMMARY

The genetic algorithm developed for finding the optimum locations for a fixed number of actuators has been demonstrated to be a powerful and time saving tool in the development of the designs of adaptive systems. The algorithm was executed on a test model for which the entire design space was known. The genetic algorithm quickly found actuator layouts with performances within the top 1% of all possible layouts. It was then demonstrated that this efficiency makes it a simple matter to perform design trades that aid in choosing the best number of actuators.

6. REFERENCES

1. *SigFit* is optomechanical post-processing and pre-processing software available from Sigmadyne, Inc., www.sigmadyne.com.
2. Goldberg, David E., **Genetic Algorithms in Search, Optimization & Machine Learning**, Addison-Wesley, Boston, 1989.
3. Doyle, Genberg, Michels, **Integrated Optomechanical Analysis**, SPIE Press, Bellingham, WA, 2002.