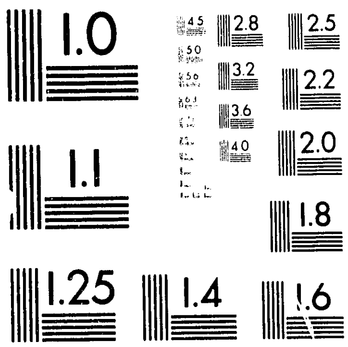
1.0
4.5
5.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

1 of 1

Object Oriented Simulation implementation in support of Robust System Design

Abstract A very brief description of two "classes" developed for use in design optimization and sensitivity analyses are given. These classes are used in simulations of systems in early design phases as well as system response assessments. The instanciated classes were coupled to system models to demonstrate the practicality and efficiency of using these objects in complex robust design processes.
keywords: Object-Oriented, Simulation, Robust design, Taguchi techniques, genetic algorithms

Introduction

Design efforts involve the selection of optimum design parameters from massive amounts of information and options. Robust design is a design paradigm which assists the engineer in finding the optimal solution in as efficient a manner as possible. Robust design has substantial history in Japanese industry and is slowly being recognized for it's potential in this country. Simply stated; robust design is a process in which designs are optimized to ensure minimal system variations to variations in materials, production, and operational environment. This design paradigm is a structured approach which will maximize the design engineers efficiency in selecting the ultimate design concept. Associated with any innovation is a cost. In the case of robust design the costs involve training and a recognition that there is an increased initial design cost. The potential payoffs of the process are a reduction in re-design efforts, a higher quality product or process, and a shorter overall time to market.

The work being reported in this paper is a description of a set of tools and an analytical approach which can assist the design engineer in his initial design efforts associated with a robust design process. The approach taken was to use the concepts of object oriented coding to develop a number of generic analytical tools for use in any design activity and a specialized set of objects for use in specific design activities. The convolution of the generic and specific object tools provides the simulation environment needed by the design engineer.

The generic analytical tools designed consist of a "Taguchi Object" and an optimization object. The Taguchi object is routinely used to assess variable or design parameter sensitivities as well as performing noise analyses on the design concepts. The optimization object is based on a number of rudimentary genetic algorithms. The optimization object is designed to handle the classical optimization problems with constraints as well as the traveling salesman problem (TSP) and multi-objective design problems.

Design Requirements

A design process must in general consider a variable number of customer objectives. The objectives include product or process cost, schedule, and the performance of a number of system parameters. The typical problem types consist of system response such as a design which optimizes circuit voltage. The second class of problem can be handled by the traveling salesman type optimization algorithm. This class of design optimization may examine trade-off's between weight and energy requirements for a space probe designed to visit vast expanses of a planetary body. The final class of optimization addresses the very real design situation in which there are a number of design objectives which must be satisfied in some optimal fashion. A design which must maximize reliability, safety, speed and ride comfort might be a typical example.

Approaching a design or analysis problem from a "robust design" perspective requires that the engineer identify the numerous variables which affect the system performance. The variables or design parameters can be classified as either control or noise parameters. The difference in the type of parameters reflect the level of control the design engineer has over the parameter. Noise parameters may represent operational temperature variations, or load variations during operation. Employing standard Taguchi techniques provide insights into which design concept will minimize the variation in system performance to these noise conditions. The technique also provides information which can be used to select an optimal set of design parameters.

The sections to follow describe the two generic objects which were developed to handle the optimization and analysis requirements associated with a robust design or analysis processes. Simple examples are included to demonstrate the utilization of these objects on aspects of current design problems.

Design Objects

The design classes were developed in C++ and constrained to interact with specifically structured system classes. The system classes had two major characteristics: first they had to be inherited from a "SYSTEM" class and second, the response functions contained no parameter list and returned a "double" variable which was a measure of system performance. These constraints were imposed because of limitations in the version of C++ being employed and on the desire to produce totally independent design classes. The system classes will be discussed in limited detail later.

Object Oriented Programming(OOP)

The problem solving paradigm associated with object oriented programming is a more natural process to a design or systems engineer than many other classical programming methodologies. The approach consists of identifying discrete elements of the problem and defining the attributes and functional characteristics of the element or object. This process results in a more natural synergism between system design and the way a human thinks and solves complex problems. In the purest sense, object oriented programming does not permit external operations to modify any of an objects attributes. Messages may be sent to an object requesting a change of state, or attribute based on internal rules of transition. As a result of this characteristic a modular, distributed development environment becomes very feasible. This represents the philosophical aspects of OOP. There are, in addition, a number of implementation characteristics which define the total OOP paradigm. The operational aspects include inheritance, polymorphism, encapsulation, and abstraction. These topics are best described in texts on OOP.

Taguchi Class

The "Taguchi class" performs three major functions associated with system design or analysis. The first function involves performing a series of designed experiments on the design or analysis problem. Specific parameters are chosen for analysis using a pre-selected orthogonal matrix as the experimental control blueprint. The Taguchi class/object illicits a system class response function assessment which sends the result to the Taguchi class for storage. Upon completion of the analytical experimentation a statistical analysis of the generated data base is performed to generate the parameter sensitivities and design space statistics. This represents the second major function of the Taguchi class. A simplified graphical representation of the Taguchi class is provided in figure 1. This representation methodology is loosely based on the "Object Modeling Technique(OMT)" developed by J. Raumbaugh et.al.

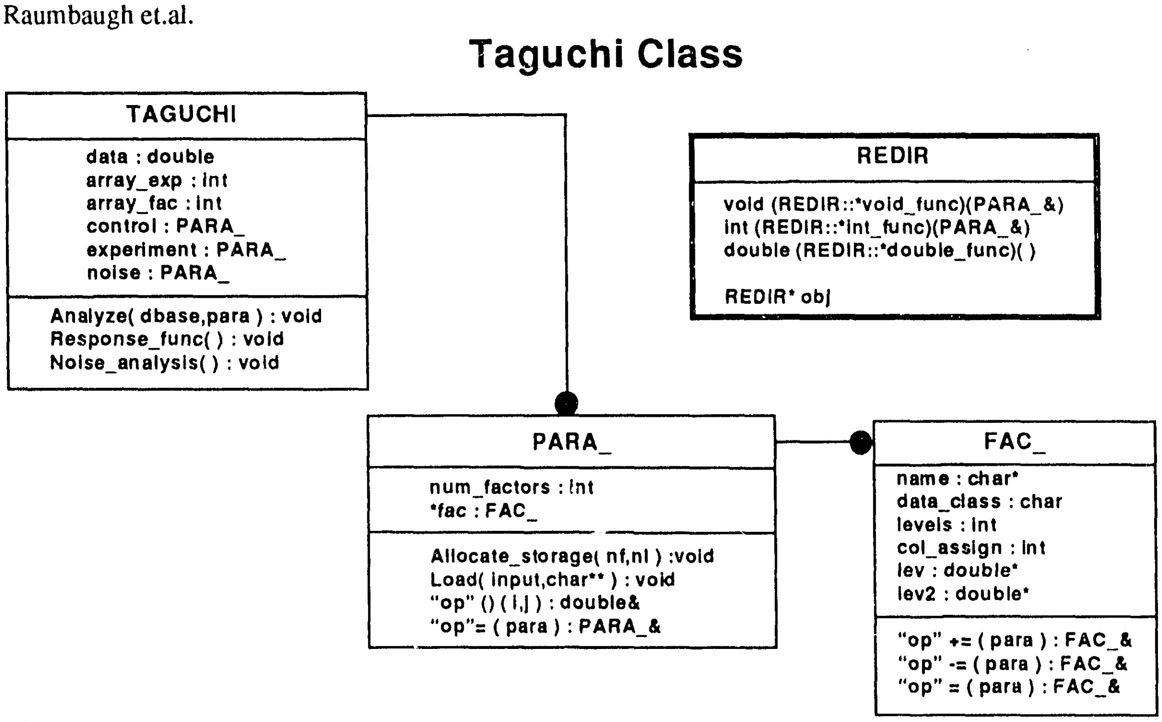

Figure 1. Taguchi and associated classes.

The Taguchi object possesses a number of PARA_ attributes which are defined as "control", "noise" and "experiment". These objects are container objects for the FAC_ objects which define the characteristics of the design variables as well as assignment information required to perform the Taguchi experiments. These objects also provide transfer mechanisms of data from the Taguchi objects to the system objects. The transfer of data is through a series of data stripping operations associated with the "SYSTEM" class. The stripping operation compares variable names found in the Taguchi PARA_ objects with model variables and if a match is found strips the data out of the Taguchi PARA_ object and replaces the system model variable with the value. This process enables the Taguchi object to run the experiments with the correct combination of design variable values or levels. The forth class in figure 1 is the REDIR class. This class provides the interface between the Taguchi class and the system classes. The declaration of this class in association with a set of masking operations makes it possible for the Taguchi object to directly access the system object response functions.

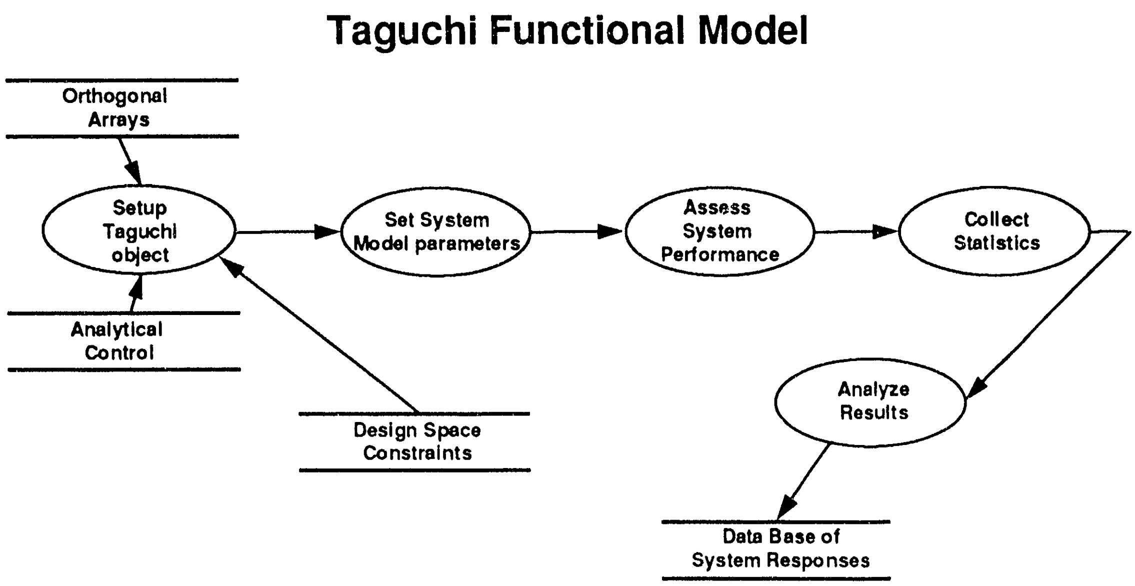

Figure 2. Simple representation of the Taguchi functional model.

The model represented in figure 2 is a very crude representation of the functional model associated with a Taguchi analysis. The setup phase of the analysis consists of locating and loading the appropriate orthogonal array, and the system design constraints. The orthogonal array selected for use in the analysis depends on the number of design factors and parameter levels to be used in the analysis. The design constraints represent the range of design options under consideration. The final block of information required consists of analytical control information. Once the Taguchi object has been defined and initialized data is stripped from the PARA_ objects and used in setting up and initializing the system models at which time a system performance calculation may be performed. The final steps consist of constructing and analyzing the statistical database generated by the Taguchi objects.

The analytical control consists of selecting a combination of design parameter levels, resetting the system model parameters and performing a system response calculation. This process is repeated for each analytical experiment performed. At the completion of this assessment the statistics can be compiled to determine the importance of the various design parameters on system performance. The assessment of noise on system performance is treated in roughly the same manner.

The third major function of the Taguchi object is to perform a noise analysis on the system model. When this option is employed a series of sub-experiments are performed to model the uncertainty associated with the design control factors as well as the noise factors. This approach provides better statistical representation of the true systems performance. Implementation of this analysis technique permits the design engineer to assess system response in a more representative environment and select design options with mitigate variations in response due to these noise conditions. This additional information provides the design engineer with options for choosing lower cost alternative design configurations if the effect of a design parameter does not effect performance beyond the assessed uncertainty limits. The "Analyze" function of the Taguchi class performs the appropriate statistical analysis representative of which ever analysis option is chosen.

Genetic Algorithm Class(GA)

The second design object being described represents the implementation of a genetic algorithm for use in optimization problems. `The genetic algorithm is a search algorithm based on the mechanics of genetics and natural selection. Roughly characterized it is a "directed random search" for an optimal solution. Classic mathematical optimization techniques often suffer from a lack of analytical robustness. A poor choice for a starting point in the search can result in the location of local maximums as opposed to the global maximum. The random facet of a genetic search increases the probability that the global maximum will be found. The principle advantages in using genetic search techniques include robustness, simplicity, and the ability to avoid converging to local maxima. A very good reference on the topic of genetic algorithms is a book by D.E. Goldberg.

The genetic algorithm, "GA_ class" performs three major functions and a number of supporting functions. The major functions consist of a functional optimization performed by the "Optimize" operation, the "traveling salesman problem" operation and a multi-criteria optimization based on the "vector maximum problem". These operations are modeled in the "TSP_optimize" and the "MCDM_optimize" routines respectively. The support functions consist of assessing genotype fitness, a scaling operation, and the control function for generating the next genotype generation. The model being described employs a genotype as opposed to the standard chromosome for generality and future extendibility. The genotype paradigm allows for more than one chromosome to represent an optimal solution. It may have application in alternate multi-criteria design problems. The "reproduce" operation provides a structure for generating a baseline "next generation" which is based on the solution fitness assessments.

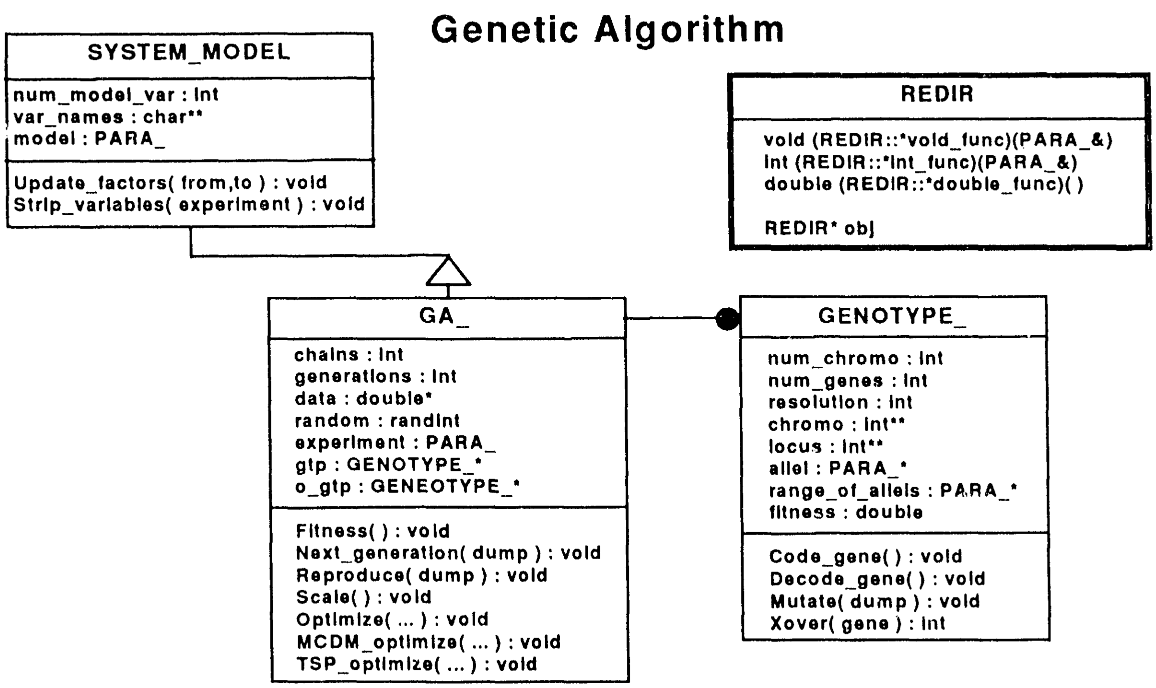

Figure 3. Classes associated with the genetic algorithm.

The "GENOTYPE_" class defines the chromosomal solution mapping, and the operations associated with genetic search. The genotype consists of an arbitrary number of chromosomes any of which maps to a total solution to a system model. Each chromosome is made up of an arbitrary number of genes, each of which represents a value for a design variable. The variable allele, and locus define the value of a gene and to location on the chromosome to which that value is coded. This modeling approach aids in the treatment of the traveling salesman problem.

The current genotype operators consist of the coding and decoding operations which correlate the chromosomal representation with the system model variables. The "Mutate "and "Xover" operations provide functional representation of the mutation operator and the chromosomal "cross-over" operator.

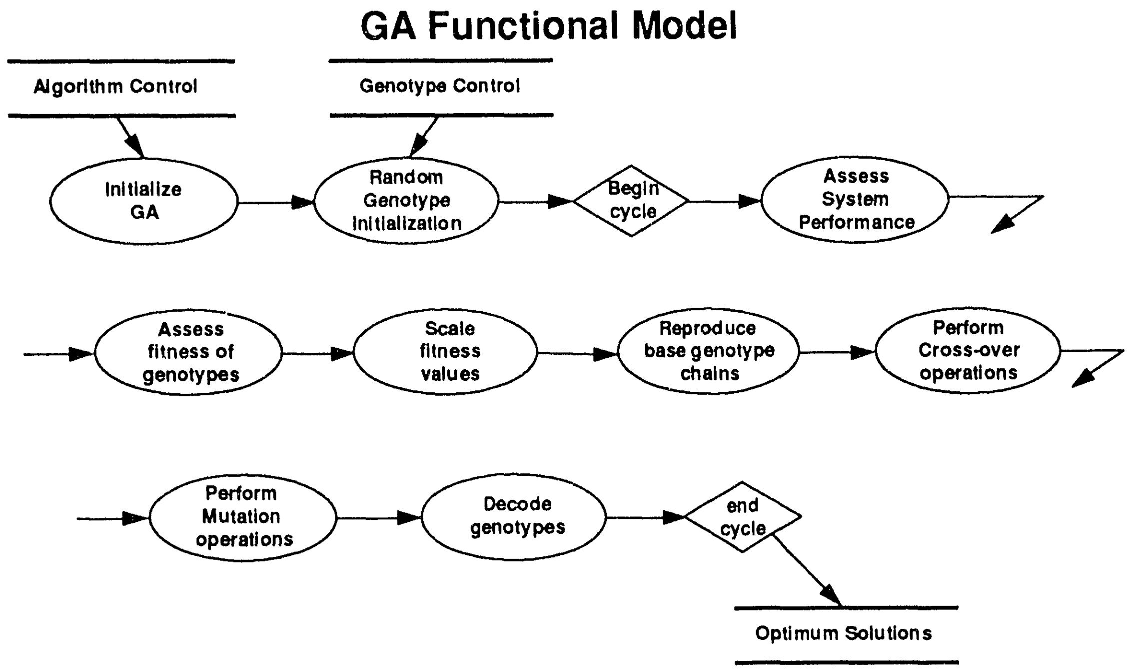

Figure 4. Functional model for the genetic algorithm.

The functional model of the genetic algorithm in figure 4 provides a simplified view of the processes involved in the search for an optimal solution to a system problem. The model represented in the figure is characteristic of the classical optimization problem the traveling salesman problem and the multi-objective problem which have been coded into the GA discussed previously. The genetic algorithm accesses the system response function, through the REDIR class, in a manner similar to the Taguchi class discussed earlier.

The first step in the search for an optimal solution consists of the random initialization of each genotype in the chain. The length of this genotype chain is controlled through user defined input values. The random initialization is intended to mitigate prolonged searches in a region of the solution space which is not likely to possess the optimal solution. The values represented by the genotype are transferred to the system model via a stripping operation similar to the operation used in the Taguchi objects. System response is defined for each genotype in the chain. The next step in the process is to assess the "fitness" of each genotype and generate a set of probabilities for use in establishing the next generation of genotypes. The scaling operation indicated in the function model is done to ensure a suitable spread in the probabilities used in the reproduction operation. With insufficient spread the rate of convergence can be diminished. The final steps in generating the next generation is to operate on the baseline genotype chains with the mutation operator and the crossover operator. Mutation consists of performing a series of bit permutations based on the user defined mutation probability. The cross-over operator selects a chromosome location on two randomly selected chromosomes of a genotype and performs a splicing operation to generate two new chromosomes.

Restricted Design Example

A sub-set of the types of design problems encountered by engineers have been included to indicate the potential these type of design objects hold. The extent of possible applications is limited only by the imagination of the engineer and his or her ability to define an appropriate system response function. Three problems have been selected for inclusion to demonstrate the range of applicability. The first problem is the more classic design problem in which we are attempting to select the be^t design configuration in order to produce the most robust design possible. The design involves an environmental sensor which must operate in very severe environments. The second problem involves a variation on the "site location" problem. Given a problem of designing a distribution system, where and how many distribution points need to be established to optimize a specific objective. The third problem involves a truck routing problem. In this situation we employ the TSP optimization algorithm in order to find the best route available to minimize the fuel requirements of the delivery system.

Sensor Design Analysis

The first problem involves a design problem which benefited from the employment of the Taguchi object in a robust design process. The environmental sensor must detect it's location relative to an obstruction plane and then "report" and act on this information prior to the system self destructing. The response function which was developed involved a time race between a successful completion of the system function and it's destruction. The time requirements for the sensor operation involved electronic, mechanical as well as physical timing issues. Associated with these timing issues was a variability or uncertainty in the values. This problem like most design problems, is a stochastic design problem not a deterministic problem as perceived by persons of other than an engineering discipline.

The problem was solved by using Taguchi noise analysis methodologies to assess the critical operational conditions at which the system should just barely function and the standard deviation of these conditions. This information was then used to construct a "probability of function" correlation for use in estimating the ability of the system to meet system performance requirements. The curves in figure 5 below were generated, again using an instanciation of the Taguchi class to assess the design and operational performance sensitivities over the total design space. The design space consisted of two different technologies, four design configurations and a range of operational utilization's for the two operational variables.

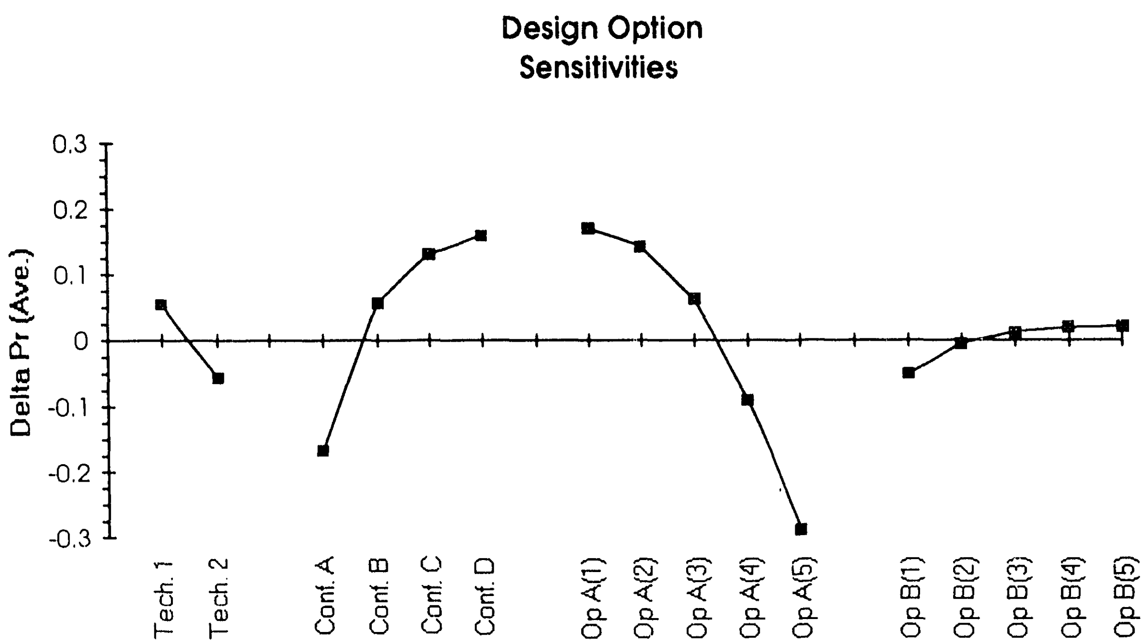

Figure 5. Results of a design sensitivity analysis using a Taguchi object.

The curves indicate substantial variation in performance due to design configuration and technology choice. The first curve shows that we might expect a change of nearly 0.1 in total probability of function in technology 1 versus technology 2. In a similar manner there is a total change in functional probability of nearly 0.35 in selecting design configuration D over configuration A. The remaining two curves demonstrate performance sensitivity to operational utilization. The implication from these curves is that system under operation A(1) and operation B(5) will result in the best system performance.

Site Location Optimization

The second problem considered involves a standard, single objective optimization problem. The problem is to select a site for a distribution system in which the only objective is to minimize the distance from the distribution site to the terminal locations. This type of problem represents a spectrum of problems ranging from a production plant design, to a missile targeting problem. The response function implemented in this situation was to maximize a function which defined the inverse sum of the distance between the distribution site to each terminal site. The simplicity of the response function was selected to provide a simple environment for "debugging" the optimization algorithm. The solution to this problem was treated through an implementation of the genetic algorithms discussed earlier.

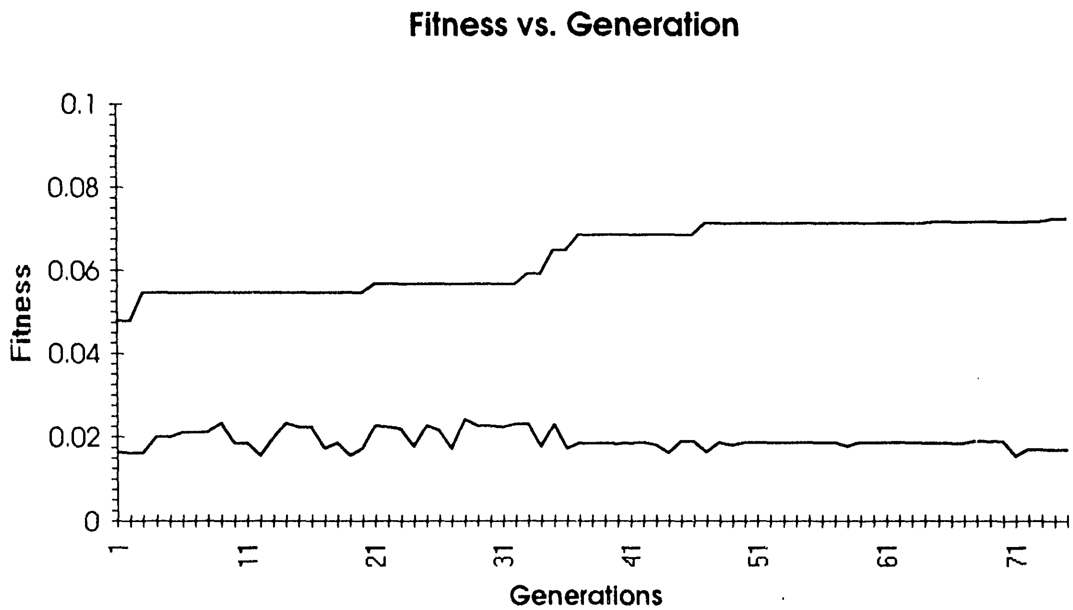

Figure 6. Fitness of the best and worst genotype as a function of generation.

The solution was obtained after 75 generations in which the probabilities of mutation and cross-over were set to 0.20 and 0.75 respectfully. The figure indicates the rate of convergence as a function of generation.

Routing Optimization

The third problem examined represents a follow on phase of the previous problem. In this situation a routing selection is being made which will minimize the total distance traveled in order to visit each of the terminal sites. This problem is a standard "traveling salesman problem" in this situation the constraint that the truck return to the distribution center was not imposed. We fired the driver at the end of the route.

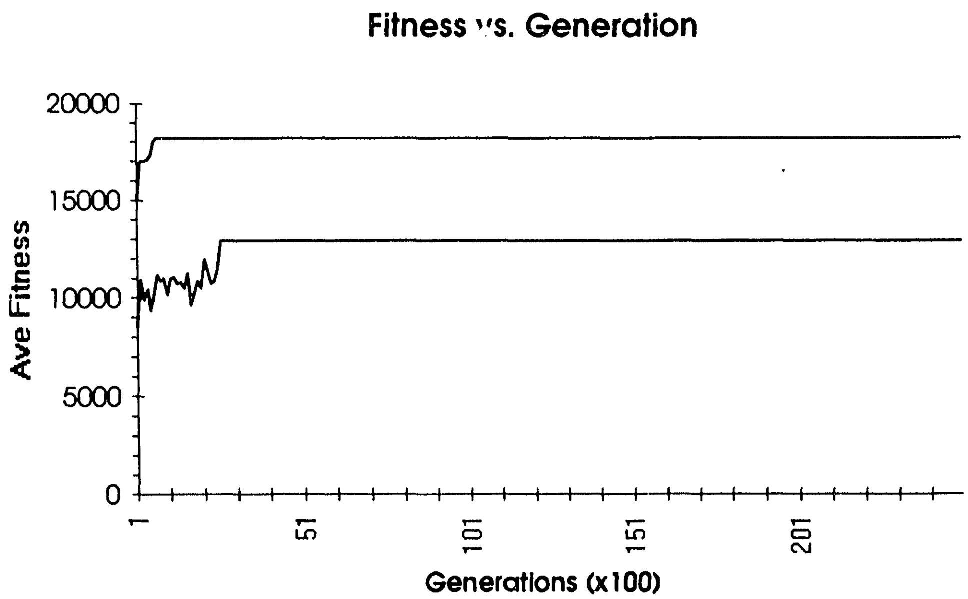

Figure 7. Fitness of the optimal route as a function of generation.

The solution was represents 2500 generations of genotypes. In this case the probability of mutation was set to 0.4 and the cross over probability was set to 0.75. The curves in figure 7 represent 100 samples per data point. The number of permutations which would have had to be performed for a twenty city problem is in excess of 2.4 $x10^{18}$. An exhaustive combinatorial search could have taken more than 50000 years of computer time, not a very practical effort.

Conclusions

The introduction of object oriented simulation tools into an engineering design environment can provide valuable insights into the expected performance of a system under realistic operational environments as well as locating optimal solutions to design and analysis problems. The examples used in this work represent initial forays into these areas of analysis but have demonstrated to the author the value and potential of further implementation. The two optimization examples could have been combined into a single multi-objective optimization problem which may have resulted in a better combined solution. The most important aspects of these implementations is the fact that classes can be developed which can be applied to many analysis and design problems providing the engineer with sufficient data to make informed design decisions.

References

M.S. Phadke; "Quality Engineering using Robust Design"; Prentice Hall, 1989; ISBN 0-13-745167-9
D.E. Goldberg; "Genetic Algorithms in Search, Optimization & Machine Learning"; Addison Wesley Publishing Co. Inc.; January 1989; ISBN 0-201-15767-5
J.E. Raumbaugh et. al.; "Object Oriented Modeling and Design"; Prentice Hall Inc.; 1991; ISBN 0-13-629841-9

DISCLAIMER

DATE
FILMED
12 / 13 / 93

END