GENETIC ALGORITHM AND GAME THEORY FOR MULTIOBJECTIVE OPTIMIZATION OF SEISMIC STRUCTURES WITH/WITHOUT CONTROL

FRANKLIN Y. CHENG and DAN LI
Department of Civil Engineering, University of Missouri-Rolla, Rolla, MO 65401

Paper No. 1503. (quote when citing this article)
Eleventh World Conference on Earthquake Engineering
ISBN: 0 08 042822 3

ABSTRACT

Genetic algorithms are both global and efficient for structural optimization. Based on genetic algorithm and cooperative game theory, new multiobjective optimization algorithms are presented in this research. The game theory method is used to develop a technique for finding a compromise solution among conflicting objectives. In the Pareto genetic algorithm, a new operator named Pareto-set filter is developed. The filter avoids missing Pareto optimal points, and makes the Pareto genetic algorithm more robust. The presented niche technique effectively prevents genetic drift influence on the evolutionary process. A seismic structure with/without control is investigated. The analysis procedure shows that the multiobjective optimization of a seismic structure which works under the environment of static and dynamic loadings is necessary to better utilize available design resource and achieve ideal dynamic response at minimal cost.

KEYWORDS

Active control; game theory; genetic algorithm; multiobjective optimization; niche technique; Pareto optimum; Pareto-set filter; seismic structure.

INTRODUCTION

In modern structural design, optimum methods play an increasingly important role. Most real-world design optimization problems in structures are multimodal. There often exist several criteria to be considered by the designer. Usually these objectives are conflicting, or competing, rather than complementary. For such a problem, a multiobjective formulation is appropriate. A single-objective optimization formulation can give an optimum design with respect to an optimal objective, but the design may not always be a "good" design. Consider a hypothetical example. If a structure is optimized for minimum cost subject to constraints on stress, displacement, buckling and period, an economical design is obtained. However, the structure may have a poor dynamic response under the action of seismic loadings. If the minimum earthquake input energy is also included as one of optimum criteria, a more rational, compromise design will be produced (Cheng *et al.*, 1996a). The latter is a multiobjective optimization problem (MOP). Multiobjective optimization offers the possibility to consider effectively all the different, mutually conflicting requirements inherent in a design problem.

Genetic algorithms (GAs), developed by Holland (Holland, 1975) based on the biological evolution mechanism and Darwinian survival-of-the-fittest theory, have been shown to be both global and efficient over a broad spectrum of problems. GAs require no gradient information and produce multiple optima rather than a single, local optimum, and can search for many solutions in parallel. In game theory, if players agree to cooperate, a Pareto optimum will be an ideal solution because it has the property that if any other solution is used, at least one player's performance index is worse, or all the players do the same. The same solution concept also applies to MOPs. This study demonstrates how game theory as a design tool applies to an MOP. The relationship between cooperative game theory and Pareto optimal solution is described. Three genetic algorithms for multiobjective optimization are proposed based on game theory. In the Pareto GA, whose goal is to find a representative sampling of solutions along the Pareto optimal set, two new techniques are investigated. A new operator called Pareto-set filter is introduced to prevent the loss of Pareto optimum points in evolutionary progress. A niche technique is created by putting limitations on reproduction operators. The proposed multiobjective optimization techniques are applied to the optimum design of a seismic structure with/without control. Numerical results show that multiobjective optimization is necessary to produce a good seismic structural design. In the example, the compromise design can lower structural weight (cost) and, at the same time, achieve ideal seismic control at minimal cost.

MULTIOBJECTIVE OPTIMIZATION AND PARETO OPTIMUM

Multiobjective optimization can be defined as determining a vector of design variables that are within the feasible region to minimize (maximize) a vector of objective functions that usually conflict with each other. It can be mathematically expressed as follows

$$\begin{aligned} &\textit{Minimize} && F(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\} \\ &\textit{Subject to} && g(x) \le 0 \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the vector of design variables, $f_i(\mathbf{x})$ is the ith objective function, and $\mathbf{g(x)}$ is the constraint vector.

One feature of multiobjective optimization is the appearance of conflicting objectives. In multiobjective optimization there exists a trade-off among objectives, i.e., an improvement gained for one objective is only achieved by making concessions to another objective. There is no optimum solution for all m objective functions simultaneously. There only exists a "compromise solution" rather than an optimum solution. Thus, the solution of a multiobjective optimization problem can be defined so that if vector $\mathbf{x}^*$ is a solution to Eq. (1), there exists no feasible vector $\mathbf{x}$ which would decrease some objective function without causing a simultaneous increase in at least one objective function. The solution following the above definition is also called a Pareto optimum or nondominated solution.

A feasible vector $\mathbf{x}^*$ is a Pareto optimum for Eq. (1), if and only if there exists no feasible vector $\mathbf{x}$ such that $f_i(\mathbf{x}) \le f_i(\mathbf{x}^*)$, $i \in \{1,2,\ldots,m\}$ and $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$, for at least one $i \in \{1,2,\ldots,m\}$ (Schdarovszky *et al.*, 1986). For a multiobjective optimization problem, the solution procedure is composed as generating its Pareto optimal set and deciding the final selection from the set.

SIMPLE GENETIC ALGORITHM

Genetic algorithm is a search procedure with a randomized, yet structured information exchange in a finite space. GA uses random processes to produce an initial population, and simple operators are applied to the population to produce a new population. The new population is referred to as offspring, and the original population as parents. This process (evolution) is repeated until a satisfactory solution evolves. GA represents complex models by simple encoding. It works with a coding of the parameter set, not the parameters themselves. One encoding method is the representation of a model by binary bit strings. Considering an optimal design problem, design variables are $\{x\}=\{x_1, x_2, x_3\}$ and each variable has a specified range so that $x_{i,min} \le x_i \le x_{i,max}$. A variable can be encoded by mapping in a range x_{min} to x_{max} using an n bit, binary unsigned integer. If a 4-bit code is chosen to mapping a variable, then $x_{min} \rightarrow (0000)$ and $x_{max} \rightarrow (1111)$. There are 2^n values in this range (for this example, n=4). The following equation can be used to decode a string.

$$x_j = x_{j,\min} + \frac{B_j}{2^n - 1}(x_{j,\max} - x_{j,\min}) \tag{2}$$

where B_j is the decimal integer value of binary string for variable x_j. A string usually stands for a point in a design variable space. In the above example, let $x_{1,min}=x_{2,min}=x_{3,min}=0$ and $x_{1,max}=x_{2,max}=x_{3,max}=4$. A design point {x}={1.8667, 2.4, 3.2} then could be represented by the string (011110011100), where the first 4 bits equal x_1 (1.8667), the middle 4 bits equal x_2 (2.4), and the last 4 bits equal x_3 (3.2).

Being different from a traditional optimization method which usually starts its search from a point $\{x\}^0=\{x1, x2, x3\}^0$, GA search works from a population of points, $\{x\}^0_1$, $\{x\}^0_2$, ..., $\{x\}^0_m$. In traditional method, a search moves gingerly from one point in the decision space to the next, using some decision rules to tell how to get to the next point. In GA, the process works generation by generation (iteration) using probabilistic transition rules, not deterministic transition rules, successively generating a new population of strings. A simple genetic algorithm is composed of the following three operators (Goldberg, 1989): reproduction, crossover, and mutation.

Reproduction is a selection process to determine the survival potential of a string according to that string's fitness. Fitness is a non-negative merit measure of optimization objective function with respect to a string. Strings with higher fitness values have a higher probability of proceeding to the next generation. For example, if fitness value of a string is f_j in a population (population size is m), the probability of the *i*th string chosen as a parent is

$$c_i=\frac{f_i}{\frac{1}{m}\sum f_j} \quad (3)$$

Crossover involves random exchange of corresponding bits between two parent strings to produce two new offspring strings. Crossover operator follows the reproduction procedure. It is a process in which all of the newly reproduced strings are grouped into pairs at random; then each pair of strings undergoes crossover based on certain rules. For example, consider a pair of parents given by previous bit string (011110011100) and another string (011001001011), representing the point (1.6, 1.0667, 2.933). If one-point crossover is used, the strings after crossover (the crossover position is selected to be 5) are (011111001011) and (011000011100), representing a pair of points (1.8667, 3.2, 2.9333) and (1.6, 0.2667, 3.2).

Mutation introduces diversity in a model population by occasional random change in bit values of strings. It plays a secondary role in the operation of genetic algorithms. In mutation operator, the value of the bit of a randomly chosen string at a randomly selected position between 1 and L (L is the length of the string) is changed. Mutation is potentially useful in restoring lost diversity in a population.

GAME THEORY ALGORITHMS

Game theory has been developed for both cooperative and non-cooperative games (Neumann *et al.*, 1947). In the latter each player acts independently in an effort to maximize his own payoff, which produces an outcome (Nash equilibrium solution) that may be favorable for one player, but poor for another. The concept of player cooperation therefore becomes important when considering compromise game outcomes. A cooperative game means that the players agree to form coalitions under the expectation that, by working together, a mutually beneficial outcome can be obtained. The measure of success of a cooperative play is embodied in the concept of Pareto optimum since it has the property of nondominated (Pareto optimum) solution. A cooperative game theory consists of analyzing conflicts existing in objectives or interest groups (players), providing an unemotional form for discussion and negotiations among players, and then suggesting a "compromise solution" which can be accepted by all players.

A multiobjective optimization problem can be cast as a cooperative game problem in which it is assumed that one player is associated with an objective. The objective function f_i can be regarded as the payoff of the *i*th player. With cooperative multiobjective optimization, the "compromise solution" should make sure that each objective obtains its maximum possible value although each objective cannot arrive at its own best value. Optimal trade-off among the objectives is sought by using the concept of game theory as follows

(Cheng *et al.*, 1996a). First, the m individual objective functions are minimized respectively subject to given constraints.

$$\begin{array}{ll} \textit{Minimize} & f_i(x) \\ \textit{Subject to} & g(x) \leq 0 \end{array} \tag{4}$$

For each objective function f_i, an optimal solution x^*_i is obtained; then a pay-off matrix is constructed as

$$[P_0]=\begin{bmatrix} f_1(x_1^*), & f_2(x_1^*), & \dots, & f_m(x_1^*) \\ f_1(x_2^*), & f_2(x_2^*), & \dots, & f_m(x_2^*) \\ & \dots & & \dots \\ f_1(x_m^*), & f_2(x_m^*), & \dots, & f_m(x_m^*) \end{bmatrix} \tag{5}$$

For each objective function, its best and worst values in the Pareto set can be obtained from the above matrix

$$\begin{array}{ll} f_{i,\min}=f_i(x_i^*) & i=1,\dots,m \\ f_{i,\max}=Max[f_i(x_j^*)] & j=1,\dots,m \quad i=1,\dots,m \end{array} \tag{6}$$

From the pay-off matrix, it can be observed that during the cooperative optimization, the *i*th objective function should not expect a value better than $f_{i,min}$, but not worse than $f_{i,max}$. Based on the above consideration, a substitute objective function can be constructed as

$$S=\prod_{i=1}^{m}\frac{[f_{i,\max}-f_i(x)]}{[f_{i,\max}-f_{i,\min}]}=\prod_{i=1}^{m}\bar{f}_i(x) \tag{7}$$

Maximizing the surrogate function S produces a solution that results in optimal compliance with multiple objectives subject to given constraints. The solution is a Pareto optimum (Cheng et al., 1996a) and stands for the "rational compromise" of the conflicting objectives. The multiobjective optimization of Eq. (1) can be transferred as

$$\begin{array}{ll} \textit{Maximize} & S=\prod_{i=1}^{m}\bar{f}_i(x) \\ \textit{Subject to} & g(x)\leq 0 \end{array} \tag{8}$$

MULTIOBJECTIVE OPTIMIZATION WITH GENETIC ALGORITHMS

GA can be used directly for unconstrained single-objective optimization problems. A fitness function connects with the objective function in a linear or nonlinear formulation. For constrained optimization problems, one approach is to transform them to unconstrained ones by penalty function methods (Richardson *et al.*, 1989). There are usually two categories of algorithms to solve multiobjective optimization problems. One is directly seeking the rational compromise solution; the other is generating the Pareto optimal set of a problem. Then the decisionmaker's preference should be determined from the set on the basis of trade-off analysis of objectives. In this section, three methods are proposed for multiobjective optimization with GAs. Belonging to the first category, the first two methods solve a multiobjective problem in a single-objective optimization pattern. They optimize the game theory Eq. (8) and give a rational compromise solution. There are mainly three disadvantages of these methods. (1) a decisionmaker cannot determine whether the given solution is steady and robust. (2) the entire procedure works in a black-box pattern. (3) there are no alternatives for further comparison. The third method is Pareto genetic algorithm which generates a Pareto optimal set. The set gives a decisionmaker insight into the trade-off change between the design objectives, and the sensitivity and robustness properties of solutions. By exploring the Pareto optimal set, a decisionmaker can select the most satisfactory solution.

Direct Method

In this method, a vector (multiobjective) optimization problem is reduced into a scalar optimization problem, where the game theory algorithm is used for the transforming procedure. When the substitute problem is constructed, a simple GA is applied to optimize it. The method consists of 4 steps: (1) use GA to optimize each objective function subject to the given constraints (Eq. 4); (2) construct the pay-off matrix $[P_o]$ (Eq. 5); (3) find the best and worst values of each objective function (Eq. 6), and construct the substitute function S (Eq. 7); and (4) use GA to optimize the scalar objective optimization problem Eq. (8).

This algorithm needs to perform m+1 scalar optimization analyses. The given optimum point from the 4th step, as mentioned, is a rational compromise which is in the Pareto optimal set.

Multiple Subpopulation Method

This method, in a similar manner, also optimizes the surrogate problem Eq. (8). However, the analytical procedure forms a parallel mathematic model. It can therefore be solved by combining genetic algorithms and parallel computer architecture. In this method, the whole population is divided into m+1 subpopulations. Each subpopulation, subject to the same constraints, has equal size but a different optimization objective. For the *i*th subpopulation ($1 \le i \le m$), its objective is f_i. For the m+1 subpopulation, its objective is the substitute function S (Eq. 7). At generation K, the pay-off matrix $[P_o]$ (Eq. 5) is constructed based on optimum values from the i to m subpopulations at generation K-1. The substitute function S at generation K is formed according to the pay-off matrix. When the optimal values of S become steady, the evolutionary process can be stopped. The outcome from the m+1 subpopulation is the optimum goal.

Pareto Genetic Algorithm

This algorithm locates the Pareto optimal set of an MOP (Eq. 1). It is constructed by using the parallel search and group output properties of GAs as well as the nondominated property of multiobjective optimization solutions. Its operating procedure is developed by revising simple GA techniques. Besides the three basic operators, reproduction, crossover and mutation, this proposed Pareto GA has two other operators: niche and Pareto-set filter.

Pareto GA deals with a vector function space. Its fitness should express the merit of a point in an objective function domain, and each point that is nondominated in a model population should be equally important and considered an optimum goal. To achieve this purpose, a nondominated rank procedure (Goldberg, 1989) is introduced. A population rank is a continuous labeling procedure. At each generation, nondominated points (strings) are picked out and assigned rank 1. From the remaining population, nondominated points are identified and assigned rank 2. This process continues for rank 3, rank 4 and so on until the entire population is ranked. In reproduction, strings of rank 1 get the most prior copies while strings with higher rank get fewer copies. When the whole population is ranked, the fitness of each string in a rank can be determined by the following equations (Cheng et al., 1996b)

$$F_i=(N_r-i+1)/SS$$

$$SS=\frac{\sum_{i=1}^{N_r}(N_r-i+1)P_{si}}{M} \qquad (9)$$

where M is population size, N_r is the highest rank of population, P_{si} is population size of rank i, and F_i expresses fitness of a string ranked i.

Niche forces individuals to share available resources and maintain appropriate diversity. Effective niche technique is the key to the success of a Pareto GA because its optimum goal is the nondominated region — Pareto optimal set. A niche technique, derived from Cavicchio's concept (Cavicchio, 1972), is constructed. After a pair of new offspring is produced, the parents are either replaced or proceed to the next generation.

Replacement occurs only when the rank of one offspring in its parent population is no worse than the best rank of their parents. Otherwise, the parents proceed to next generation.

Pareto-set filter reduces effects of genetic drift, and makes a Pareto GA more robust. Reproduction of a new generation cannot guarantee that the best traits of the parents are always inherited by their offspring. Some of these traits may never appear in a future phase of evolution due to limited population size. In evolutionary processes, many points appear once or twice and then disappear forever. Some of them may happen to be the sought-for optimum goals, i.e., the Pareto optimum points. To stop the loss of Pareto optimal points, a new concept called Pareto-set filter is introduced. A Pareto-set filter pools nondominated points ranked 1 at each generation, and drops dominated points. At each generation, points ranked 1 are put into a filter. When new points are added to the Pareto-set filter, all points in the filter are subjected to a nondominated check (filtering process) and dominated points are discharged. Thus, only nondominated points are stored in the filter. Size of a filter can equal any reasonable value. When the population size of points in a filter surpasses a given value, the points at a minimum distance relative to other points are removed to maintain an even distribution in the filter.

For an MOP, the analysis procedure for the proposed Pareto GA is composed as follows: (Cheng *et al.*, 1996b, c) (1) obtain an evenly distributed subset of Pareto optimal set, and (2) choose a reasonable solution from the set. The decision procedure to choose a compromise solution from a Pareto optimal set is constructed by the Pareto-set filter with the cooperative game theory. At each generation, the maximum and minimum values of each objective are found from the filter. Then the substitute function S (Eq. 7) is constructed. Checking each group of functions in the filter, the group which makes S maximum value produces the optimum solution for the current generation. If maximum S keeps constant, continuing for 10 or more generations, it can be considered the convergence of the evolutionary processes and the analysis is ended. The points in the Pareto-set filter are on or close to the Pareto optimal set — trade-offs among objectives.

OPTIMIZATION DESIGN OF SEISMIC STRUCTURE WITH/WITHOUT CONTROL

Consider a three-story steel shear frame with an active control system located on the top floor (Cheng et al. 1996a) shown in Fig. 1. Floor diaphragms are rigid and axial deformations are neglected. Thus the system has only one degree of freedom (in the lateral direction) at each floor. The total of live and dead loads at each story is 56kN/m, which does not include the weight of the columns. Lateral forces as well as weight density ρ and elastic module E of the structure are also represented in Fig. 1.

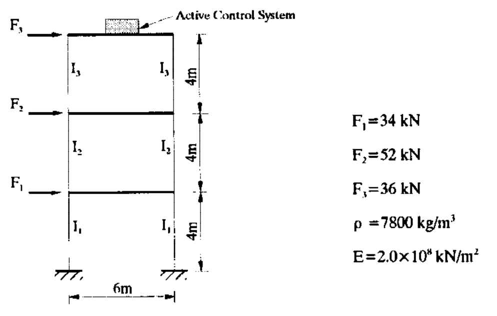

Fig. 1. Three-Story Steel Frame

Optimum design of the system takes place under two cases. First case is optimizing the system without control. Optimization criteria are structural weight W in order to better utilize materials and reduce structural cost, and earthquake input energy E_i for the purpose of reducing dynamic response and damage to the structure under the action of seismic loadings. Input energy E_i (Cheng et al., 1996a) represents the work done by the structure's base shear as the structure moves through ground displacements. Second case is optimizing the system with an active control system. Optimization criteria are the weight W and the performance index PI (Cheng et al., 1996a) to optimize control energy input. The two multiobjective optimization problems are formed as

$$\text{Case I:} \quad Min. \ (W, E_i); \qquad \text{Case II:} \quad Min. \ (W, PI) \tag{10}$$

Both cases are subject to the same constraints as

$$\begin{aligned} &|\sigma_i| \le 165{,}000 \ kN/m^2 \\ &|\delta_i| \le \frac{1}{400} h_i \\ &T_1 \ge 0.3 \\ &0.00001 \le I_i \le 0.002 \ m^4 \\ &0.5 \le \frac{K_{i+1}}{K_i} \le 1 \end{aligned} \tag{11}$$

where σ_i, δ_i and K_i are column stress, relative story displacement and story stiffness of the ith story, respectively, and i=1,2,3. T_1 is the fundamental natural period of the structure. Details of the integrated structural/control optimization design are given in Cheng's paper (Cheng et al., 1996a). The proposed genetic algorithms are applied to make optimum analysis. Due to space limitation, only the results from the Pareto GA are presented and discussed. GA parameters are that uniform crossover (Davis, 1991) and stochastic remainder selection (Goldberg, 1989) are used in crossover and selection operating procedures. Crossover probability is 0.60, and mutation probability is 0.01. Chromosome length is 60. Population size for case I and II is 300 and 400, respectively.

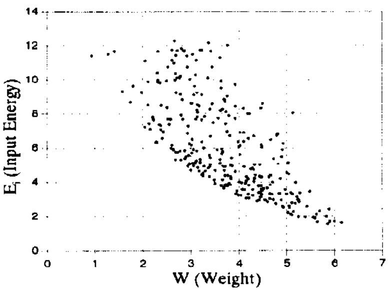

Fig. 2. Initial Population Distribution (Generation 0)

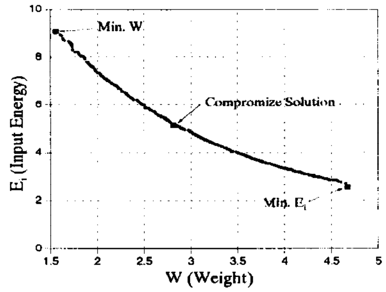

Fig. 3. Points in Pareto-Set Filter (Generation 100)

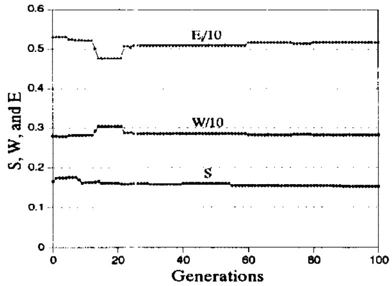

Fig. 4. Compromise Solutions

Figures 2 through 4 illustrate part of the analysis procedure for case I. Initial population distribution is displayed in Fig. 2. They are far from the optimum goal — the Pareto optimal region — and most of them are infeasible points. Figure 3 shows the points in the Pareto-set filter which gives the trade-offs between the two conflicting objectives. The two points (Min. W and Min E_i), determined by using a traditional optimal method, are drawn as reference points because they are the two end points of the Pareto optimal set for this problem. One can say that the points in the filter are on or very close to the Pareto optimal set. The compromise solution point (W=2.8110, E_i=5.1635) is the decision-making result. Figure 4 gives the compromise solution in accord with the evolutionary progress. At each generation, the magnitude of S (Eq. 7) relates to the minimum and maximum values of each objective function and distribution of points in the filter. When the population uniformly converges to a Pareto optimal set, the S will become steady.

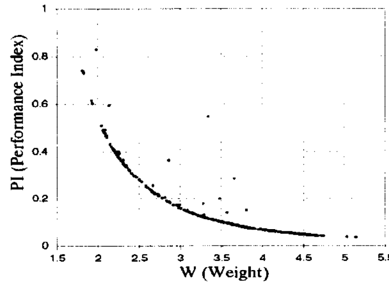

Fig.5. Population Distribution (Generation 200)

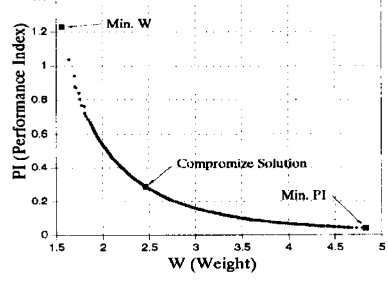

Fig. 6. Points in Pareto-Set Filter (Generation 200)

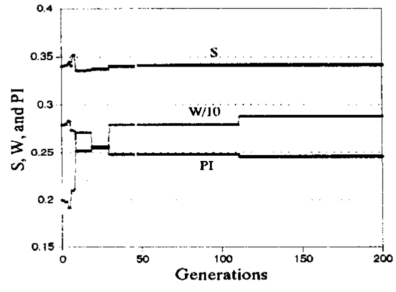

Fig. 7. Compromise Solutions

Results of case II are given in Figs. 5 through 7. Figure 5 shows the population distribution in generation 200. Comparing Fig. 5 and Fig. 6, it can be observed that points from the Pareto-set filter represent a much better nondominated set than do points of rank 1 (Fig. 6). Filter operator significantly improves the solution of a Pareto GA.

CONCLUSIONS

In this paper, the concepts of genetic algorithms, game theory and Pareto optimal set are presented. Based on them, new algorithms for multiobjective optimization are developed. The Pareto GA is demonstrated on a seismic structure with/without control. Seismic structures are subject to static and dynamic loading action. A single-objective optimization usually cannot yield a rational optimal design. Therefore, a multiobjective formulation is appropriate.

Multiobjective optimization provides a designer with a systematic method for considering all the conflicting design requirements inherent in an application. Game theory defines a decision method for guiding an optimum process and, as such, results in optimal compliance with multiple objectives. Comparing traditional multiobjective optimization methods, Pareto GA is an effective numerical algorithm for generating Pareto optima. The Pareto-set filter enables the Pareto GA to be more robust, stable and computationally feasible for even extremely difficult problems. The niche technique effectively prevents genetic drift. It involves no new parameters, and the computational cost for niche operating is negligible.

ACKNOWLEDGMENTS

This presentation is a portion of research results from projects partially supported by the National Science Foundation under grants MSS-9214664 and BCS-9302201 as well as Intelligent Systems Center at UMR. The financial support is gratefully acknowledged.

REFERENCES

Cavicchio, D. J. (1972). Reproductive Adaptive Plans. *Proc. ACM 1972 Annual Conference,* Boston, 1-11.

Cheng, F. Y. and D. Li (1996a) Multiobjective Optimization of Structures with/without Control. *J. Guidance, Control, and Dynamics*, Vol. 19, No. 2.

Cheng, F. Y. and D. Li (1996b). Genetic Algorithm Development for Multiobjective Optimization. *National Science Foundation Report*, January 1996.

Cheng, F. Y. and D. Li (1996c). Constrained Multiobjective Optimization Design with Pareto Genetic Algorithms, submitted to *J. Struct. Div., ASCE*, January 1996.

Davis, L. (ed.) (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York.

Goldberg, D. E. and M. P. Samtani (1986). Engineering Optimization Via Genetic Algorithm. *Proc. 9th Conference on Electronic Computation*, ASCE, New York, 471-482.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, Mich.

Neumann, V. and Morgenstern O. (1947). *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, N. J.

Richardson, J. T., M. R. Palmer, G. Liepins and M. Hilliard (1989). Some Guidelines for Genetic Algorithms with Penalty Functions. *Proc. 3rd International Conference on Genetic Algorithms*, San Mateo, Calif. 191-197.