ELSEVIER

Comput. Methods Appl. Mech. Engrg. 163 (1998) 87–94

Computer methods in applied mechanics and engineering

Application of genetic algorithms to the design optimization of an active vehicle suspension system

A.E. Baumal, J.J. McPhee*, P.H. Calamai

Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Received 9 October 1996

Abstract

The use of numerical optimization methods to partially automate the design process is demonstrated. Genetic algorithm (GA) optimization, a global search technique, is used to determine both the active control and passive mechanical parameters of a vehicle suspension system. The objective is to minimize the extreme acceleration of the passenger's seat, subject to constraints representing the required road-holding ability and suspension working space. GA optimization is also used for passive suspension design to compare results from the literature based on a local optimization search technique: a gradient projection method.

1. Introduction

Multibody dynamics is used extensively by industry to model and design vehicle systems and sub-systems, including automobiles, trains and the suspensions for both. From a design perspective, the drawback of most commercial codes is that they only provide analyses of systems whose parameters have been specified. Design optimization, parametric studies and sensitivity analyses are difficult, if not impossible to perform [1]. Instead, design engineers must decide on how to change parameter values and re-perform the analysis until a set of performance measures becomes acceptable. This 'manual' process, often accompanied by prototype testing, can be difficult and time-consuming for complex systems with nonlinear performance measures. In addition, active elements can introduce behaviour in suspension systems that is not intuitive [2]. Numerical optimization helps automate the design process by altering parameter values in a search to minimize an objective function subject to constraints, which may reflect performance characteristics. Fig. 1 illustrates this semi-automated design process that requires the system and optimization statement models as inputs. In practice, an engineer interprets the resulting design and, depending upon its suitability, either constructs a prototype or re-formulates the optimization problem. In order to determine the potential contribution of optimization methods to the design process, a global optimization technique was used to design an active vehicle suspension.

Optimal control theory has predominantly been used in the past to determine feedback gains for active suspension systems [3,4]. Numerical optimization methods have also been applied to passive suspension design where the mechanical parameters (spring and damping values) are the design variables [5]. In this paper, we consider the active control and passive mechanical parameters concurrently as design variables, as recently suggested by Bestle [2] and Schiehlen [6]. The specific problem considered is the passive half-car model and road profile of Haug and Arora [5], with the addition of active components for comparison. Others use this same system for passive design to test various optimization techniques with differing specified suspension parameters, performance criteria and road surfaces [7–10]. We used a genetic algorithm, a stochastic global optimization

* Corresponding author.

0045-7825/98/$19.00
PII: S0045-7825(98)00004-8

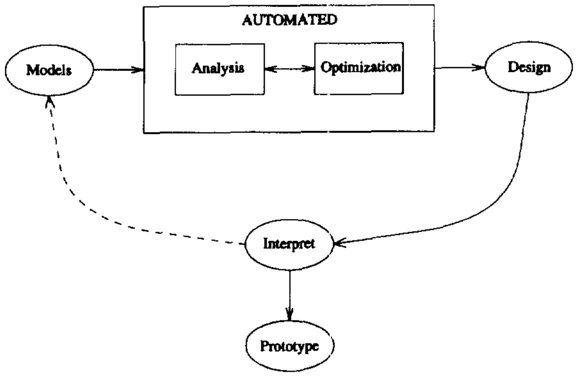

Fig. 1. The semi-automated design process.

technique that is based on a mathematical model of the natural process: survival of the fittest design [11]. Previous methods for the passive system are of the local search type, some using zeroeth-order techniques, while others are first-order nonlinear programming methods requiring gradients of the objective function and constraints with respect to the design variables. Five aspects of the semi-automated design process are emphasized by first defining the suspension *model*, then stating the *optimization* problem and genetic algorithm procedure, highlighting the issues regarding the *analysis*, and finally presenting *design* results and their possible *interpretations*.

2. Mathematical model

The following is a description of the model used by Haug and Arora [5], but with active components included and all parameter values converted to SI units. Fig. 2 illustrates the road vehicle suspension system considered for optimal design. The system's 5 degrees of freedom are represented by the independent generalized co-ordinates

$$\{q\} = \{q_1, q_2, q_3, q_4, q_5\}^{\mathrm{T}} \quad (1)$$

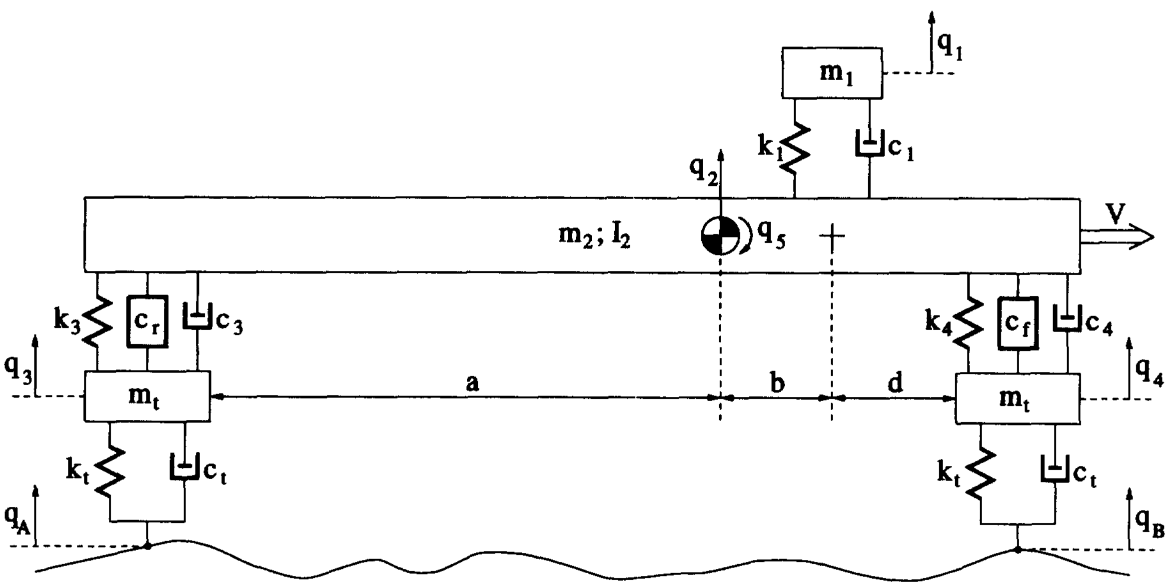

Fig. 2. Five degrees of freedom half-car model.

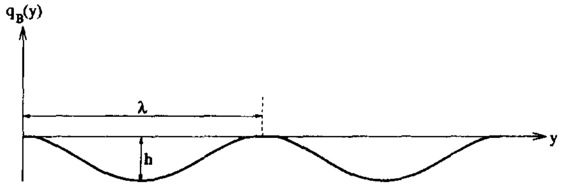

Fig. 3. The road surface profile.

which are measured from the static equilibrium position. A mathematical description of the model is expressed by the equations of motion

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{f(t)\} \tag{2}$$

where $[M]$, $[C]$ and $[K]$ are the mass, damping and stiffness matrices, respectively.

The mass properties of the system are defined by: the seat mass $m_1 = 132$ kg, the car-body mass $m_2 = 2040$ kg and its moment of inertia $I_2 = 4630\ \text{kg}\cdot\text{m}^2$, and the wheel mass $m_t = 44$ kg. It is assumed that all dampers and springs behave linearly and that the rotation q_5 is small enough so that the linear form of (2) is valid. Two active elements provide forces proportional to the absolute vertical velocity of the points on the car body directly above the rear and front wheels. These devices, characterized by proportionality constants c_r and c_f, are known as skyhook dampers and are more effective in reducing car-body motions than passive dampers [6]. The forcing function $\{f(t)\}$ depends on the spring-damper model of the tire, with $k_t = 262700$ N/m and $c_t = 876\ \text{N}\cdot\text{s/m}$, and the road disturbance, q_A and q_B. Fig. 3 shows the sinusoidal shape of the road profile consisting of two successive depressions of depth $h = 0.102$ m and length λ. Two separate profiles, of lengths $\lambda_1 = 24.4$ m and $\lambda_2 = 3.05$ m, are considered.

As functions of time, the road conditions are given by

$$q_B(t) = \begin{cases} \frac{h}{2}(\cos(\omega t) - 1)\,, & \text{if } 0 \leqslant t \leqslant \frac{2\lambda}{V} \\ 0\,, & \text{otherwise} \end{cases} \tag{3}$$

and

$$q_A(t) = \begin{cases} \frac{h}{2}(\cos(\omega(t-\tau)) - 1)\,, & \text{if } \tau \leqslant t \leqslant \left(\tau + \frac{2\lambda}{V}\right) \\ 0\,, & \text{otherwise} \end{cases} \tag{4}$$

where τ and ω, the time lag between wheels and the forcing frequency, respectively, are given by

$$\tau = \frac{a+b+d}{V} \tag{5}$$

$$\omega = \frac{2\pi V}{\lambda} \tag{6}$$

with dimensions $a = 2.03$ m, $b = 0.25$ m and $d = 0.76$ m, and vehicle velocity $V = 24.4$ m/s. The remaining eight unknown parameters comprise the set of design variables:

$$\{x\} = \{k_1, c_1, k_3, c_3, c_r, k_4, c_4, c_f\}^{\mathrm{T}}\,. \tag{7}$$

3. Optimization and analysis

Analysis of the suspension generally implies solving (2) for the time response of the system. To help identify issues regarding the analysis, such as the time interval for numerical integration, the optimization problem and procedure is stated first.

3.1. Optimization design statement

There are three characteristics commonly used to assess the performance of vehicle suspension systems [3–6]:

(i) *ride comfort*, which improves as the magnitude of the seat acceleration, $|\ddot{q}_1|$, is reduced.

(ii) *road-holding ability or safety*, which is acceptable for restricted or low tire-road contact forces and is quantified by tire deflection (e.g. the rear tire deflection $q_3 - q_A$).

(iii) *the suspension working space*, which must be restricted (e.g. the rear wheel-chassis space $q_2 + aq_5 - q_3$).

Ride comfort is chosen to be the most important characteristic so it is expressed in an objective function as

$$\min f(\{x\}) = \max|\ddot{q}_1(t)|_i\,, \quad t \in [0, T]\,, \quad i = 1, 2 \tag{8}$$

where the index i refers to the two different road profiles; thus, the system response to both profiles must be calculated. Note that an upper bound is placed on the maximum seat acceleration:

$$g_1 = f - 10.3\ \mathrm{m/s^2} \leq 0 \tag{9}$$

in agreement with Haug and Arora [5]. The other two performance characteristics are included as further constraints so that the tire deflections and relative spaces between bodies are restricted by

$$g_2 = |q_4(t) - q_B(t)| - 0.0508\ \mathrm{m} \leq 0 \tag{10}$$

$$g_3 = |q_3(t) - q_A(t)| - 0.0508\ \mathrm{m} \leq 0 \tag{11}$$

$$g_4 = |q_1(t) - q_2(t) + bq_5(t)| - 0.0508\ \mathrm{m} \leq 0 \tag{12}$$

$$g_5 = |q_2(t) + aq_5(t) - q_3(t)| - 0.127\ \mathrm{m} \leq 0 \tag{13}$$

$$g_6 = |q_2(t) - (b + d)q_5(t) - q_4(t)| - 0.127\ \mathrm{m} \leq 0\,. \tag{14}$$

The constraints (10) through (14) must hold for all times $t \in [0, T]$ and for both road conditions. The design variables are also limited to ranges defined by the bounds shown in Table 1. While the bounds for all these constraints have been converted to SI from [5], the active skyhook dampers have been chosen to take on values ten times that of the passive dampers to further inhibit absolute motion of the car body. In practice, these ranges should reflect component availability and the financial and technical abilities to manufacture them.

3.2. Genetic algorithm optimization procedure

Goldberg [12] and Haftka and Gürdal [13] describe genetic algorithms and their applications to engineering design. The genetic algorithm (GA) for optimization starts from an initial set, or first generation, of randomly chosen designs with uniform probability distribution. The population of each generation will have feasible design variables in terms of their allowable ranges (i.e. Table 1) but may be infeasible otherwise. Each design is represented by a finite-length binary string that consists of smaller strings that decode to a value for each design variable. The eight design variables' strings are taken to have 6 binary sites so that there are $2^6 = 64$ possibilities for each variable and $(2^6)^8 = 2.8 \times 10^{14}$ possible designs. That is, the design variables only take on values from a discrete set. Given a current generation of designs, or strings, there are three steps used to implement the algorithm:

(i) *Reproduction*, which is performed by copying a current generation string into a new population, the

Table 1
Design variable ranges

Design variables	Lower bound	Upper bound
k_1 [N/m]	8756	87563
c_1 [N·s/m]	350	8756
k_3, k_4 [N/m]	35025	175127
c_3, c_4 [N·s/m]	875	14010
c_r, c_f [N·s/m]	8756	140101

'parent pool', according to the design's fitness. The fitness must be evaluated for each design in this generation and it depends on the objective function value, f, and constraint violations. The GA searches for designs with high or maximum fitness, while f is to be minimized. The fitness, Y, that represents a minimizing objective is

$$Y = f_{\max} - \alpha f - G_p \tag{15}$$

where G_p is a penalty when constraints are violated

$$G_p = \eta \sum_{i=1}^{6} \max(0, g_i) . \tag{16}$$

It is important to ensure a wide range of fitness values so that small changes in the design string result in significantly different Y values. To achieve this, it was found by trial and error that the values 1000, 100 and 10 000 for $f_{\max}$, α and η, respectively, scale the fitness well. In practice, one often chooses $f_{\max}$, with $\alpha = 1$, to be the largest value of f obtainable from a design that is feasible for all constraints; however, the design variables for this situation are too difficult to find for the given design statement. To create the parent pool, designs are copied with a bias toward strings with higher Y. A way to select such designs is to 'spin' a weighted roulette wheel where each string occupies an area that is proportional to its fitness value relative to the total fitness sum of the entire generation. Since designs are copied and not removed from the roulette wheel, the parent pool may contain multiple copies of designs with high fitness.

(ii) *Crossover*, which is the exchange of design characteristics among randomly selected pairs from the parent pool. This is done by randomly choosing a portion along the design string length that identifies the pattern of 0's and 1's that gets exchanged. Here, one-point crossover is used such that only one position along the string identifies this portion. For example, two strings of length 5

parent 1: 0 1||1 1 1

parent 2: 1 1||0 1 0

mate at the crossover point 2 to create

offspring 1: 0 1||0 1 0

offspring 2: 1 1||1 1 1 .

(iii) *Mutation*, which is achieved by switching a 0 with a 1, or vice-versa, at a binary site. The reproduction step can copy many of the same designs, which, if chosen in pairs for crossover, causes no change to the population. This third step diversifies the population so that different areas of the design space can be explored. The mutation probability per site was taken to be 0.2%.

The population passed from one step to the other remains the same size. It was found by trial and error, as with the other GA parameters such as mutation probability, that a population size of 100 gives fairly consistent results. A complete iteration or new generation of designs is formed after completing all three steps. The algorithm stops when the maximum fitness design comprises at least 30% of a newly created generation. The GA uses stochastic ideas based on analogies with the natural process. For example, the design string with binary sites is analogous to the chromosome with genes. The reproduction stage itself is a simulation of the survival of the fittest designs.

The simple GA as given by Goldberg [12] was implemented in a program written in C using pseudo-random number generators linked from the Numerical Algorithms Group (NAG) Fortran library [14]. However, instead of using a weighted roulette wheel based on the fitness *sum* of the population for the reproduction stage, one based on the *ranking* of individuals in the population according to fitness is used, as described by Whitley [15]. In addition, in order to improve the efficiency of the GA, the binary strings and fitness values for each unique design of the current generation are stored in a linear search look-up table [16]. If a design string in the next generation matches one in the table, then the fitness does not have to be re-calculated. This can save computing time, especially for expensive fitness evaluations, as is the case here, since a complete dynamic analysis of the suspension system must be performed for each set of design variables for both road profiles.

3.3. Dynamic analysis

No gradient information for GA optimization is required; only evaluations of the objective function, f, and the constraints, g_i, are necessary to determine fitness. In order to evaluate these, the equations of motion, (2), need to be numerically integrated for $t \in [0, T]$. All initial conditions are zero. The final time T must be chosen carefully so that extreme accelerations caused by many possible designs are included. However, if it is chosen too large, the integration time increases and slows down the GA which requires many analyses. Here, T was selected to be 5 s as suggested by Paeng and Arora [9]. In addition, the type of numerical integration affects the analysis time. It was found that using the single-step 4th-order Runge–Kutta technique [17] every 0.0025 s (for a total of 2000 time steps) gives very accurate results. The GA optimization, dynamic analysis and fitness evaluation were linked and automated on a Silicon Graphics Indigo 2 XZ workstation.

4. Design results and interpretations

Table 2 shows values of the design variables for five independent runs of the GA (i.e. the pseudo-random number generator was seeded by the computer clock so that different results may be obtained for each run). Also shown are the peak absolute acceleration values of the seat for the vehicle subjected to both road profiles where the higher of the two for each design is the value of the objective function, f, and is highlighted in the table. The number of generations and analyses (or fitness evaluations) and CPU time required to converge are also given. The seat suspension values, k_1 and c_1, are consistently at or near their lower bounds, providing a 'soft suspension', possibly keeping the forces applied to the seat, and hence its acceleration, low. Both wheel passive dampers, c_3 and c_4, are also always at their lower bounds, perhaps in an effort to let the active skyhook dampers more effectively reduce car-body motions. The best design, as measured by the lowest objective function value, was obtained from run 5, which required 2558 analyses—only 9.1×10^{-10}% of the total number of possible designs. Fig. 4 illustrates the convergence of run 5 by plotting the objective function value, f, that corresponds to the design with highest fitness, Y, in each generation versus the generation number.

The GA was also used for passive suspension design to compare results from those obtained by Haug and Arora [5] who used a gradient projection method, a local search optimization technique. Table 3 displays the results and Fig. 5 compares the seat acceleration responses (for the road profile causing the maximum peak value) of the GA passive and best GA active designs. The active system's response shows that the road

Table 2
GA results for active suspension design

	Run 1	Run 2	Run 3	Run 4	Run 5
k_1 [N/m]	8756	8756	8756	12510	12510
c_1 [N·s/m]	484	484	484	350	350
k_3 [N/m]	123979	117307	88397	68383	108412
c_3 [N·s/m]	875	875	1293	875	875
c_r [N·s/m]	140101	133847	98405	79641	123423
k_4 [N/m]	57264	55040	57264	72831	63935
c_4 [N·s/m]	875	875	875	875	875
c_f [N·s/m]	106744	106744	112999	140101	123423
max $\lvert\ddot{q}_1(t)\rvert_1$ [m/s^2]	**0.3850**	**0.3738**	0.3748	0.3650	0.3662
max $\lvert\ddot{q}_1(t)\rvert_{12}$ [m/s^2]	0.3786	0.3714	**0.3829**	**0.3706**	**0.3705**
# of generations	53	47	38	59	57
# of analyses	3185	2605	2223	2660	2558
CPU time [s]	860	700	595	715	687

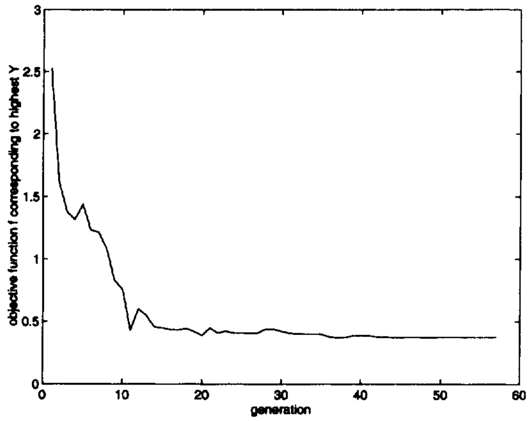

Fig. 4. Convergence of GA run 5.

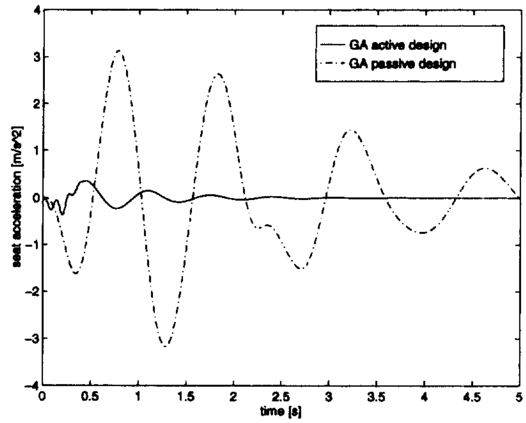

Fig. 5. Acceleration response for the passive and active designs.

Table 3
Comparison of passive designs

	Gradient projection	GA (search table)	GA (no table)
k_1 [N/m]	8756	63796	63796
c_1 [N·s/m]	1564	751	751
k_3 [N/m]	35025	72831	72831
c_3 [N·s/m]	6622	3169	3169
k_4 [N/m]	35025	35025	35025
c_4 [N·s/m]	8042	1918	1918
$\max \lvert \ddot{q}_1(t) \rvert_1$ [m/s^2]	3.19	**3.17**	**3.17**
$\max \lvert \ddot{q}_1(t) \rvert_2$ [m/s^2]	**3.30**	3.12	3.12
iterations	40	25	25
# of analyses	40	1436	2500
CPU time [s]	NA	321	630

disturbance has little effect on the seat acceleration. The GA passive design, while not significantly better than the one found by the gradient projection method, does demonstrate that there exists other feasible designs with lower objective function values. Although the GA requires many more analyses than local nonlinear optimization methods, it does not require sensitivity analyses to determine the gradients of the objective function and constraints with respect to the set of design variables. In particular, determining the gradient of point-wise constraints (constraints that must hold over an entire time interval) may cause instability in executing nonlinear programming methods and can be an involved computation requiring the solution of additional ordinary differential equations [18].

Tables 2 and 3 also demonstrate the benefit of using a linear search look-up table to avoid re-computing the fitness for previously analyzed designs. Conventional implementations of the GA, without a search table, perform a number of analyses equal to the number of generations multiplied by the population size, as demonstrated in the last column of Table 3. With the search table implementation however, Table 3 shows that the same results can be obtained (the random number generator was seeded with the same value) in almost half the number of analyses and CPU time. The benefit gained in using a look-up table depends on the CPU time required to perform a fitness evaluation as well as GA processing determined by such parameters as population size, design string length, probability of mutation, the stopping criteria and the pseudo-random number generator seed. As an example, run 5 of Table 2, while requiring more generations to converge than runs 1 and 2, required fewer analyses and hence less CPU time.

5. Conclusions

Genetic algorithm (GA) optimization, a global technique, searches for a design that minimizes an objective function subject to constraints. As an example of using numerical optimization to help automate the design process, GA optimization was used to determine parameter values for an active vehicle suspension that minimized a performance criteria while satisfying a number of other design requirements. The response of the active design that best minimizes the objective function shows that the road disturbance has little effect on the seat acceleration when compared to passive designs. In practice, however, realistic implementation of the active skyhook dampers must be addressed by obtaining actuators that can deliver the required power, if possible.

To compare results with a local optimization search technique, the gradient projection method [5], the GA was also used for the design of the passive suspension system. The GA results illustrate that other feasible designs exist with lower objective function values than those found with local optimization search methods. Although the GA requires more computing effort than a gradient projection method, it does not require gradients of the objective function and constraints with respect to the set of design variables.

It was also shown that the efficiency of the GA can be improved by monitoring previously analyzed designs so as to avoid re-computing the fitness for a same design. To further improve efficiency and consistency in results, the GA parameter values, such as population size and mutation probability, may be tuned more effectively [13]. Most importantly, the genetic algorithm results show that there is potential to incorporate global optimization methods for suspension system design. Other global techniques [11], which may utilize local nonlinear programming (such as a gradient projection method) to help refine the results, should be examined.

Acknowledgment

The financial support of this research by Bombardier Inc. and the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

[1] W. Kortüm, W. Schiehlen and M. Hoffmann, Progress in integrated system analysis and design software for controlled vehicles, Supplement to Vehicle Syst. Dyn. 23 (1994) 274–296.

[2] D. Bestle, Optimization of automotive systems, in: E.J. Haug, ed., Concurrent Engineering: Tools and Technologies for Mechanical System Design (Springer-Verlag, Berlin, 1993) 671–683.

[3] R.S. Sharp and D.A. Crolla, Road vehicle suspension system design—A review, Vehicle Syst. Dyn. 16 (1987) 167–192.

[4] A.G. Thompson, An active suspension with optimal linear state feedback, Vehicle Syst. Dyn. 5 (1976) 187–203.

[5] E.J. Haug and J.S. Arora, Applied Optimal Design: Mechanical and Structural Systems (John Wiley and Sons, New York, 1979).

[6] W. Schiehlen, Symbolic computations in multibody systems, in: M. Pereira and J. Ambrósio, eds., Computer-Aided Analysis of Rigid and Flexible Mechanical Systems (Kluwer Academic Publishers, Dordrecht, 1994) 101–136.

[7] E.J. Haug, N.K. Mani and P. Krishnaswami, Design sensitivity analysis and optimization of dynamically driven systems, in: E.J. Haug, ed., Computer Aided Analysis and Optimization of Mechanical System Dynamics (Springer-Verlag, Berlin, 1984) 555–632.

[8] J. Missaoui, M.J. Richard and L. Cheng, An extension of the vector-network model to spacial dynamic simulation and design of constrained mechanical systems, Eur. J. Mech. A/Solids 13 (1994) 459–473.

[9] J.K. Paeng and J.S. Arora, Dynamic response optimization of mechanical systems with multiplier methods, J. Mech. Trans. Autom. Des. 111 (1989) 73–80.

[10] L.F.P. Etman, D.H. Van Campen and A.J.G. Schoofs, Optimization of multibody systems using approximation concepts, in: D. Bestle and W. Schiehlen, eds., IUTAM Symposium on Optimization of Mechanical Systems (Kluwer Academic Publishers, Dordrecht, 1996).

[11] J.S. Arora, O.A. Elwakeil and A.I. Chahande, Global optimization methods for engineering applications: A review, Struct. Optim. 9 (1995) 137–159.

[12] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning (Addison-Wesley, Reading, Massachusetts, 1989).

[13] R.T. Haftka and Z. Gürdal, Elements of Structural Optimization, 3rd rev. and expanded ed. (Kluwer Academic Publishers, Dordrecht, 1993).

[14] Numerical Algorithms Group, The NAG Fortran Library Introductory Guide, Mark 15 (NAG Ltd., Oxford, UK, 1991).

[15] D. Whitley, The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, in: J.D. Schaffer, ed., Proc. Third Int. Conf. on Genetic Algorithms (Morgan Kaufmann Publishers, San Mateo, California, 1989) 116–121.

[16] R. Sedgewick, Algorithms in C (Addison-Wesley, Reading, Massachusetts, 1990).

[17] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing (Cambridge University Press, Cambridge, 1986).

[18] C.C. Hsieh and J.S. Arora, A hybrid formulation for treatment of point-wise state variable constraints in dynamic response optimization, Comput. Methods Appl. Mech. Engrg. 48 (1985) 171–189.