Application of Genetic Algorithms To Vehicle Suspension Design

HongBiao Yu
Mechanical Engineering Dept.
The Pennsylvania State University
University Park, PA 16802

Nan Yu
Mechanical Engineering Dept.
The Pennsylvania State University
University Park, PA 16802

Abstract

The primary function of a suspension system of a vehicle is to isolate the road excitations experienced by the tires from being transmitted to the passengers. In this project report, we formulate an optimal vehicle suspension design problem with the quarter-car vehicle dynamic model. The two objectives of the optimization are: 1. Minimize the maximum bouncing acceleration of the sprung mass 2. Minimize average suspension displacement subject to a number of constraints. The constraints arise from the practical kinetic and comfortability considerations, such as limits of the maximum vertical acceleration of the sprung mass and the suspension working space. In solving this problem, the genetic algorithms have consistently found near-optimal solutions within specified parameters ranges for several independent runs. For validation, the solution obtained by GA was compared to the classical suspension configuration by Haug and Arora [Haug, 1984], [Arora, 1989] and was found to yield similar performance measures. This encourages us to extend the application of GA to other more complicated vehicle dynamics problems with full confidence.

1 INTRODUCTION

1.1 RIDE COMFORTABILITY

Ride comfortability refers to the sensation of a passenger in the environment of a moving vehicle. Ride comfort problem is mainly caused by the vibrations of the vehicle body, which may be induced by a variety of sources, such as road surface irregularities, aerodynamics forces, vibrations of the engine and driveline, and non-uniformity of the tire/wheel assembly. Usually, road surface irregularities, ranging from potholes to random variations of the surface elevation profile, act as a major source that excites the vibration of the vehicle body through the tire/wheel assembly and the suspension system [Wong, 2001].

Generally speaking, human beings feel uncomfortable when exposed to vibrations with frequencies in motion sickness regime: 0.1-1Hz. And, ride comfortability is considered to be improved as the magnitude of the seat acceleration and displacement is reduced [Baumal and McPhee, 1998].

1.2 SUSPENSION DESIGN

1.2.1 overview

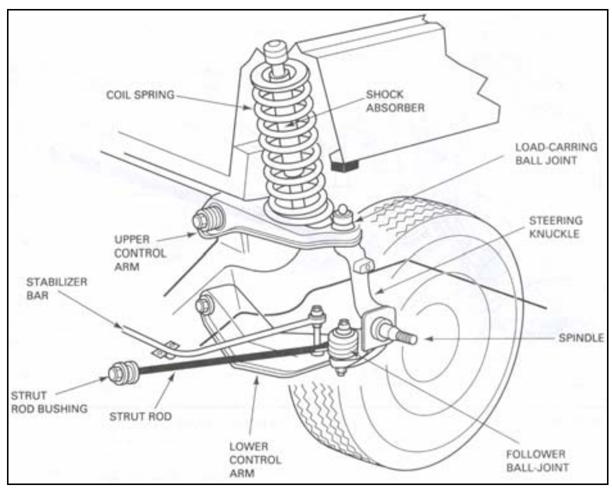

Figure 1: Suspension system

Figure 1 shows a typical independent suspension of a car. Suspension system is primarily used to isolate the road excitations from being transmitted directly to the passengers and to keep the tire-road contact. When considering ride comfort only, a suspension design involves selecting the right dynamic characteristic and geometry configuration for the suspension to minimize the seat acceleration and displacement subject to constraints, such as natural frequency of the sprung mass, suspension working space, and a number of others.

1.2.2 Design criteria

Time domain statistics, such as mean suspension deflection, maximum and RMS values of suspension acceleration are often used in suspension design as criteria for ride comfortability.

1.2.3 An important trade-off

There are a number of important trade-offs in the design of conventional suspensions. The one concerning ride comfort is between suspension displacement and acceleration. A hard configuration with high spring stiffness and high damping is required for reducing suspension displacement while a soft configuration with low spring stiffness and low damping is required for reducing suspension acceleration.

1.3 GA APPLICATION

Multi-body dynamics has been used extensively by automotive industry to model and design vehicle suspensions. Before modern optimization methods was introduced, when conducting an "optimization" on a design, engineers must first change the values of parameters and then re-perform the whole analysis again until a set of performance measures became acceptable. Design optimization, parametric studies and sensitivity analyses were difficult, if not impossible to perform. This 'manual' process usually accompanied by prototype testing, could be difficult and time-consuming for complete systems with nonlinear performance measures. In addition, many elements can introduce behaviors into the suspension systems that are not intuitive. With the development of various optimization methods, numerical optimization helps automate the design process by altering parameter values in a search to minimize/maximize an objective function subject to the constraints, which reflect some practical considerations on performance characteristics [Baumal and McPhee, 1998]. In this project, we used genetic algorithm, a stochastic global optimization technique based on mathematical models of the natural process: survival of the fittest to design a vehicle suspension. Figure 2 illustrates this semi-automated design process with the system model and optimization statement as inputs.

Four aspects of this GA design process are emphasized:

1. Building the suspension model,
2. Stating the genetic algorithm procedure and optimization problem formulation,
3. Highlighting the issues regarding the analysis,
4. Presenting design results and their possible interpretations.

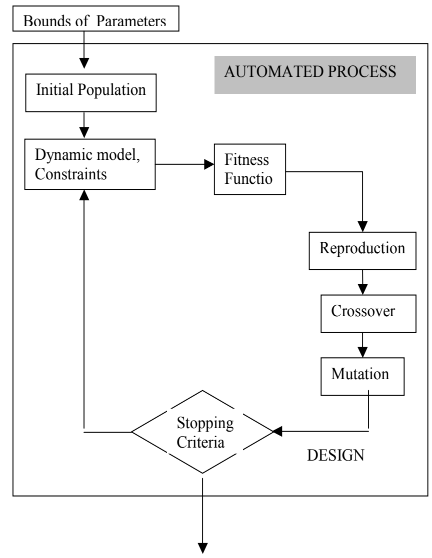

Figure 2: Design process using GA

2 MATHEMATICAL MODEL FOR SIMULATING VEHICLE DYNAMICS

2.1 QUARTER-CAR MODEL

The mathematical model used for simulating vehicle dynamic is known as "quarter car". It is traditionally used for analysis of the ride dynamics of passenger cars. In the quarter-car model, the effects of vehicle roll are assumed to be negligible. Cole and Cebon concluded from the results of validation of their three-dimensional truck model that the roll mode is not sufficiently excited on highways to contribute significantly to vehicle dynamics [Cebon, 1999]. The effects of pitch are considered by increasing the sprung mass by 100% from its original value such that the vertical motion caused by vehicle pitch (the mass moment of inertia) is incorporated.

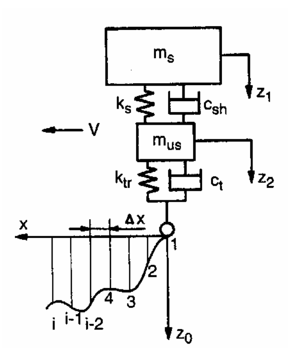

Figure 3: Quarter-car model (Wong, 2001)

A schematic of the quarter-car model is shown above in Figure 3. This model consists of one sprung mass M_s and one unsprung mass M_u. Both the sprung and the unsprung masses are considered to be rigid and are constrained to move vertically with the displacements z_1 and z_2, respectively. Therefore, the quarter-car model is a two degree-of-freedom system with the vertical displacements z_1 and z_2. The road profile elevation z_g is the input to the system. The spring-damper system with stiffness K_t and damping coefficient C_t represents the linear model of tire, which has constant point contact with the road.

2.2 EQUATIONS OF MOTIONS

The motions of the two degree-of-freedom quarter-car model are governed by the equations (2-1). This group of equations is solved by numerical time integration with respect to time (Runge-Kutta).

$$[M]\{\ddot{z}\}+[C]\{\dot{z}\}+[K]\{z\}=\{F\} \qquad (2\text{-}1)$$

The fixed parameters of the system are defined in Table 1. It is assumed that all dampers and springs behave linearly. The forcing function *{F}* depends on the spring-damper model of the tire, and the road disturbance, z_g, which will be discussed in section **2.3**.

Table 1: Given parameters of the quarter-car model

M_s	1500 kg
M_{us}	50 kg
k_t	200,000 N/m
C_t	876 N.s/m

2.3 ROAD PROFILE

The experimental road profile data used in the simulation is shown in Figure 4. This profile was sampled by UMTRI in 1989-1996 on interstate PA42 with a sampling interval of 0.152meter. As determined by the sampling interval, the maximum wave number presented in the road profiles is 3.3cycles/meter. The profile is 500 meters in length, with an IRI of 170in/mile. This value represents the average road condition defined by FHWA.

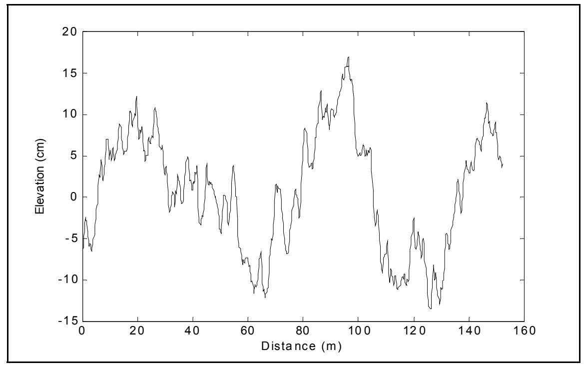

Figure 4: Road Profile

3 GENETIC ALGORITHMS AND MATLAB GA TOOLBOX

3.1 GENETIC ALGORITHMS

A brief introduction to the genetic algorithms is given in this section. Interested readers are encouraged to refer to *Handbook of Evolutionary Computation* [various authors, IOP Publishing Ltd. 1998-2000] for a more comprehensive description of GA along with other evolutionary algorithms.

Genetic algorithms (GAs) are stochastic global search and optimization methods that mimic the metaphor of natural biological evolution and which are developed based on the Darwinian theory of 'survival of fittest'. GAs differ significantly from most classic optimization techniques in many aspects. First of all, unlike classic methods, genetic algorithms are not gradient-based, i.e. they do not require the objective functions to be continuous, neither do they need information about the derivatives of the objective functions, therefore they can handle problems with discrete solution spaces. Secondly, the search mechanism is stochastic in nature, which makes them capable of searching the entire solution space with more likelihood of finding the global optima. Third, GAs are able to solve problems with non-convex solution space, where classic procedures usually fail. Fourth, genetic algorithms explore the entire space to search for the optimal solutions from a population of solutions to another population of solutions, rather than from one solution to another, this characteristic makes GAs uniquely suited to multi-objective optimization. All these differences make GAs superior over classic methods in some real-world applications, particularly for some very complex engineering problems, for example, complex truss-beam design, components design, and structure design.

In genetic algorithms, the design variables are coded as finite-length strings. Canonical GAs use binary representation: a string has a finite length and each bit of a string can be either 1 or 0. Real coding was introduced into the variations of canonical GAs. Each string is evaluated by the objective function and assigned a value, *fitness*, which determines whether or not a string to be selected for reproduction.

Simple GA has three basic operators:

Selection
Crossover
Mutation

A genetic algorithm starts iteration with an initial population. Each member in this population is evaluated and assigned a fitness value. In the *selection* procedure, some selection criterion is applied to select a certain number of strings, namely *parents,* from this population according to their fitness values. Strings with higher fitness values have more opportunities to be selected for reproduction in next step. Next, in *crossover* procedure, selected strings from old population are randomly paired to mate. For binary coding, a cross-site is determined according to some law, and the paired strings exchange all characters following the cross-site. Crossover usually results in two new strings, namely, two *children* that are expected to combine the best characters of their parents. An example of single point crossover is given.

parent 1 : 1 1 1 ¦ 1 1 1
parent 2 : 0 0 0 ¦ 0 0 0

↓ Crossover

child 1: 1 1 1 ¦ 0 0 0
child 2: 0 0 0 ¦ 1 1 1

Mutation simply changes one bit 0 to 1 and vice versa, at a position determined by some rules. Mutation is simple but still important in evolution because it further increases the diversity of the population members and enables the optimization to get out of local optima.

After mutation, a new generation is created, and thus becomes the *parents* for next generation. This process is iterated until convergence is achieved or a near optimal solution is found.

3.2 MATLAB GENETIC ALGORITHM TOOLBOX -GAOT

Since the complex vehicle dynamics model has already been developed using MATLAB, MATLAB GA toolbox was adopted in this project, though there are many advanced genetic algorithms written in C and FORTRAN. The MATLAB toolbox, GAOT(Genetic Algorithm Optimization Toolbox) was written by Houck *et al* [Houck et al 1995] in the North Carolina State University in 1995. This toolbox has both binary and real representations. Users may choose whichever suited for his problem.

3.2.1 Selection method

In this study, selection method is a ranking selection based on the normalized geometric distribution. The solutions were mapped to an ordered set, and then ranked with a value Pi. Normalized geometric ranking [Joines and Houck 1994] defines Pi for each individual by:

P [selecting the ith individual] = $q'(1-q)^{r-1}$

Where:

q = the probability of selecting the best individual
r = the rank of the individual, where 1 is the best
P = the population size

$$q' = \frac{q}{1-(1-q)^P}$$

3.2.2 Crossover

Arithmetic crossover method for real-coded representation was implemented in this study. The crossover process is described as follows. For real-coded parent X and Y, arithmetic crossover produces two complimentary linear combinations of the parents X' and Y':

$$X' = rX + (1-r)Y$$
$$Y' = (1-r)X + rY$$

Where:
r = U (0,1)
And
U (0, 1) is a uniform distribution over (0, 1).

3.2.3 Mutation

Non-uniform mutation function was chosen. It randomly selects one variable x_i, and sets it equal to a non-uniform random number. It is defined in the following equations:

$$x'_i = \begin{Bmatrix} x_i + (b_i - x_i) f(G) & \text{if } r_1 < 0.5; \\ x_i - (x_i + a_i) f(G) & \text{if } r_1 \geq 0.5; \\ x_i, & \text{otherwise} \end{Bmatrix}$$

Where,

$$f(G) \;=\; (r_2(1 - \frac{G}{G_{\max}}))^b$$

r_1, r_2 = a uniform random number between (0,1)
G = the current generation
G_{max} = the maximum number of generation
b = a shape parameter
a_i , b_i = the lower and upper bound for each variable x_i

3.2.4 Termination

As far as termination is concerned, one can either specify a maximum generation number, or a convergence threshold. The latter criterion was adopted in this project. The run is stopped when both the variation between best fitness and mean fitness and the standard deviation of one generation are less than the set tolerance. The pseudo-code is like the following:

```
If
        (best fitness - mean fitness)<=tolerance
    AND
        standard deviation <=tolerance
Then
    STOP
```

3.3 OPTIMIZATION PROCEDURES

First, genetic algorithms initialize suspension design variables Ks and Cs. Then Ks and Cs are passed into the quarter car model to solve for the dynamic response (displacement and accelerations values) of the system. These values are then substituted back into the GA process to calculate the fitness of the suspension design. This procedure is repeated until the stopping criterion is met.

4 OPTIMIZATION PROBLEM FORMULATION

4.1 OVERVIEW

The two design variables need to be optimized are K_s and C_s, which represent the stiffness and damping coefficients of the suspension, respectively. Initialized from the given parameter ranges by genetic algorithm, the values of K_s and C_s, are then evolved generation by generation and substituted into equations of motions (2-1) to solve for the time response of the system. The evolution stops when the optimal solution is obtained.

4.2 OBJECTIVES

Assessing the comfortability of a vehicle suspension system may be very complex. The two commonly adopted criteria are listed below:

Absolute Magnitude of the sprung mass acceleration, $|\ddot{z}_1|$

Absolute Magnitude of the average sprung mass displacement, $|z_1|$

Generally, ride comfortability of a suspension system is improved as $|\ddot{z}_1|$ and $|z_1|$ are both reduced. However, these two objectives are competing in nature under normal operating conditions. A suspension system satisfying one of two ride comfort criteria does not necessarily follow that the other one is also satisfied. That means, at a given point on the road surface, small displacement amplitude $|z_1|$ does not guarantee a small acceleration $|\ddot{z}_1|$. In order to properly compromises these two conflicting objectives and limited by the capability of MATLAB GA toolbox, we converted this multi-objective problem into a single-objective problem by summing up $(|z_1|+|\ddot{z}_1|)$ and require value of this converted objective to be as small as possible, that is, to minimize $(|z_1|+|\ddot{z}_1|)$.

4.3 CONSTRAINTS HANDLING

Other requirements regarding ride comfortability and kinetic properties of the suspension are treated as constraints in our problem formulation.Three constraints are specified.

The maximum amplitude of sprung mass acceleration should not exceed 1g (9.8m/s^2). This constraint is incorporated into the original objective function by adding a penalty term $\max(|\ddot{z}_1|-9.8,0)$. The fitness will be penalized by $(|\ddot{z}_1|-9.8)$ if the constraint is violated.

According to ISO2631, human beings feel motion sick when subjected to a vibration at the natural frequency of less than 1Hz. A good suspension design should imply a natural frequency of greater than 1 Hz. This constraint is accounted for by adding a penalty term $\max(1-\omega,0)$. The fitness will be penalized by $1-\omega$ if the natural frequency ω of the sprung mass is less than 1Hz.

Suspension working space refers to the absolute value of the relative displacement between the sprung and unsprung masses i.e. $|z_1-z_2|$. This value cannot be arbitrarily large from a practical point of view. It is restricted to be less than 13cm. The penalty term in the form of $\max(|z_1 - z_2| - 0.13, 0)$ will be added to the original objective function.

In summary, the problem formulation can be stated as following:

Minimize $\quad (|z_1| + |\ddot{z}_1|)$

Subject to constraints:

$$|\ddot{z}_1| < 9.8\, m/s^2$$
$$\omega > 1 \text{ Hz}$$
$$|z_1 - z_2| < 0.13 \text{ m}$$

Taking all the above statements into consideration, the objective function becomes

$$\begin{aligned}\min f(Ks, Cs) = (|z_1| + |\ddot{z}_1|) \\ + \max(|\ddot{z}_1| - 9.8, 0) + \max(\omega - 1, 0) \\ + \max(|z_1 - z_2| - 0.13, 0)\end{aligned}$$

4.4 A MODIFIED OBJECTIVE FUNCTION

The possible ranges of the suspension spring stiffness *Ks* suspension damping coefficient *Cs* were found to be [40000, 170000] N/m, and [10000, 140000] N*s/m, respectively [Haug, 1984], [Arora, 1989]. Within these parameter ranges, the maximum sprung mass acceleration $|\ddot{z}_1|$ is less than 13 m/s^2, and maximum displacement of the sprung mass $|z_1|$ is less than 0.3m.

As can been observed, the varying spaces of the two design variables are quite large, while the results space of the objective function is very small. So, the value change of the objective function may not be easily observable when Ks and/or Cs are varying. In order to ensure a wide range of fitness values so that small changes in the design variables result in significantly different objective values, we introduce five amplifying factors $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ into the original objective function to enlarge the objective space

$$\begin{aligned}&\min f(Ks, Cs) = \\ &(|z_1| + \alpha_1 |\ddot{z}_1|)\alpha_2 + \max(1 - \omega, 0)\alpha_3 \\ &+ (\max(|\ddot{z}_1| - 9.8, 0) + \alpha_4 \max(|z_1 - z_2| - 0.13, 0))\alpha_5\end{aligned}$$

α_i 's are determined by trial and error. $\alpha_1 = 3$, $\alpha_2 = 100$, $\alpha_3 = 100$, $\alpha_4 = 1000$, $\alpha_5 = 10000$.

5 DESIGN RESULTS AND INTERPRETATIONS

5.1 INITIAL POPULATION AND FINAL SOLUTION

Figure 5 shows the forced response ($|z_1|$ and $|\ddot{z}_1|$) of the suspension system of 100 initial designs and the optimal design (star). Noting that as highlighted in Sec. 4.4, the response space is quite tight for $|z_1|$ and $|\ddot{z}_1|$, the seemingly small difference between initial solutions and optimal solution actually corresponds to quite large difference in design parameters (Ks and Cs). As observed, the two objectives $|z_1|$ and $|\ddot{z}_1|$ are conflicting with each other.

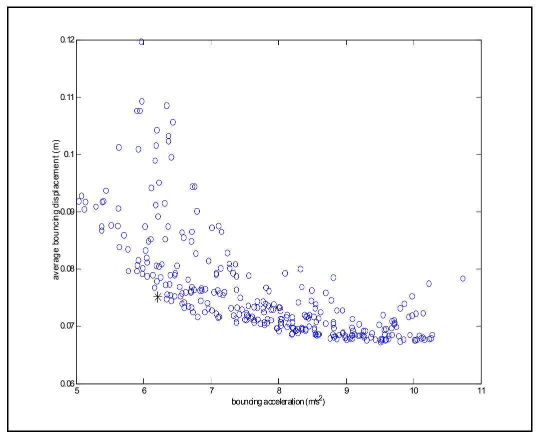

Figure 5: Initial population and Final optimal result

5.2 COMPUTATIONAL COMPLEXITY

The numbers of generations, evaluations of objective function and CPU time required for four separate runs are given in Table 2. For this specific problem, the least computational complexity occurs when the population size is set to be 100.

Table 2: Statistics for computational complexity

	Run 1	Run 2	Run 3	Run 4
Population Size	100	200	300	300
Number of Generations	22	16	16	13
Number of Evaluations	2200	3200	4800	3900
CPU time	3,033 sec	3,160 sec	6,066 sec	5,380 sec

5.3 CONVERGENCE PROCESS

Figure 6 illustrates the convergence of the three independent runs with different population size (maximum generation set to be 100) by plotting the standard deviation of objective function value in each generation against the generation number. As can be seen, in all the three runs, the objective function converged to the optimal value very fast. The convergence process of the simulation with population size set to be 300 and maximum generation number equals to 300 is shown Figure 7. As can be seen, the increased population size and generation number from 100 to 300 did not affect the convergence to the optimal result for this specific problem.

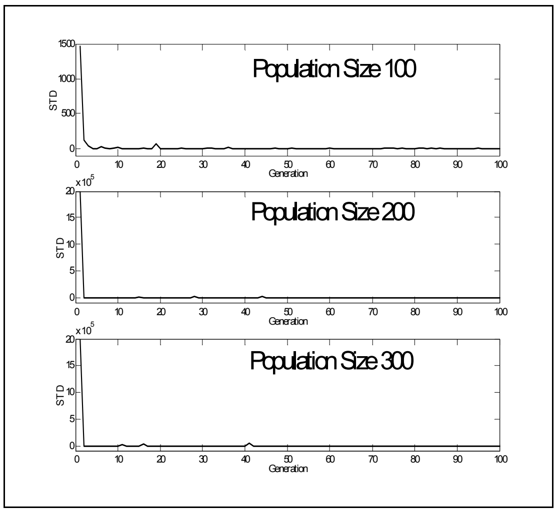

Figure 6: Convergence process for maximum generation equals to 100

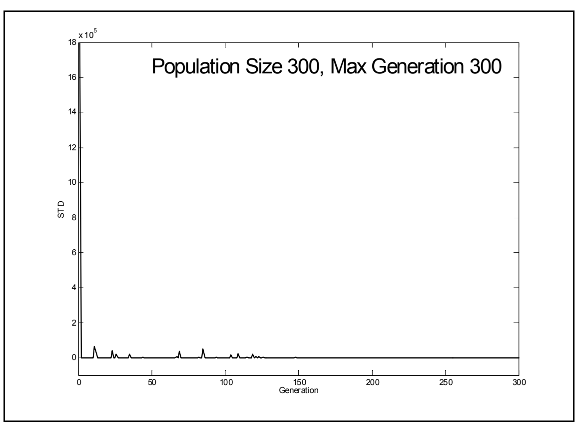

Figure 7: Convergence process for maximum generation equals to 300

5.4 DESIGN RESULTS

Table 3 shows the design results from four independent runs of the GA program. Also shown are the values of the accessing criteria: peak absolute acceleration and average displacement of the sprung mass. One can see that the four independent runs have found very similar results. The suspension parameters, k_s and c_s, are consistently near the middle point between the upper and lower bounds. This set of parameters provides a suspension that is both "soft" enough to keep the sprung mass acceleration as low as possible and "hard" enough to effectively reduce the body bouncing motions.

The results comply with our intuitive sense and can be illustrated by Figures 8 and 9. As can be seen, in comparison to the optimal design: a harder suspension with k_s*=120,000 N/m,* c_s*=80,000 N/m.s* results in a higher acceleration but smaller displacement for the sprung mass, a softer suspension with k_s*=120,000 N/m,* c_s*=80,000 N/m.s* results in a lower acceleration but larger displacement for the sprung mass. So, the final optimal design has to be a trade-off between the two conflict objectives.

Table 3: Optimal results from 4 different runs

	Run 1	Run 2	Run 3	Run 4
Population Size	100	200	300	300
Max.	100	100	100	300
K_s (N/m)	71744	72041	71719	72116
C_s (N.s/m)	48774	48752	48699	48593
Max $\left\|\ddot{z}_1\right\|$ (m/s^2)	6.3	6.3	6.3	6.3
Mean $\|z_1\|$ (m)	0.075	0.075	0.075	0.075

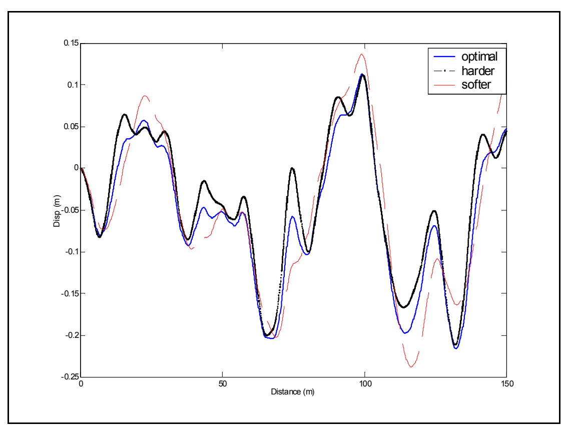

Figure 8: Comparison of displacement with other configurations.

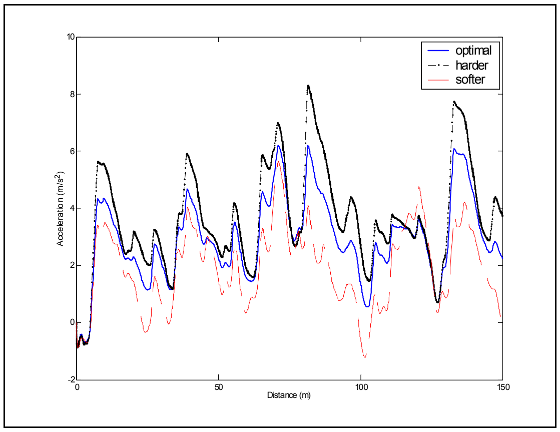

Figure 9: Comparison of acceleration with other configurations

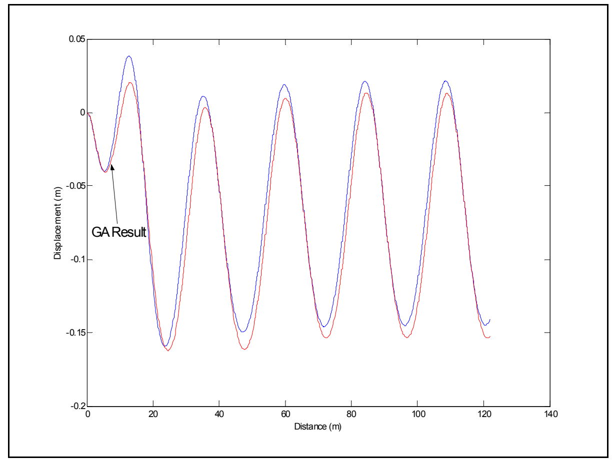

Figure 10: Comparison of displacement

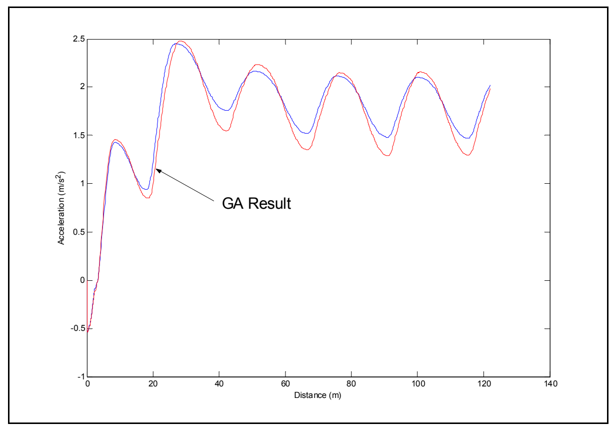

Figure 11: Comparison of acceleration

5.5 COMPARISON TO OTHER OPTIMIZATION METHOD

In order to verify the validity of the results, the GA results were compared to those obtained by Haug and Arora [Haug, 1984], [Arora, 1989] who used a gradient projection method, a local search optimization technique. To make the results comparable, the road profile and tire parameters were selected to be the same as in Haug and Arora's simulation.

The road profile used in the comparison simulation is a sine wave with a wavelength of 24.4m and an elevation of 5.1cm. The vehicle speed is set to be 24.4m/s such that resonance can be observed if the natural frequency of the suspension system is near 1Hz.

Table 4 and Figures 11-12 display the comparison results and related parameters. As can be seen, GA yields similar performance measures. This validates the GA results and also demonstrates that there exists other feasible design, which is able to achieve the same objectives.

Table 4: Comparison to results from gradient projection

	Gradient projection	Genetic Algorithm
K_t (N/m)	200,000	200,000
C_t (N.s/m)	850	850
K_s (N/m)	35025	71744
C_s (N.s/m)	80000	48774
Max $\lvert\ddot{z}_1\rvert$ (m/s2)	3.30	3.65
Mean $\lvert z_1\rvert$ (m)	0.051	0.049
Convergence	40 evaluations	2200 evaluations

6 CONCLUSIONS

Genetic algorithm optimization is a global optimization technique, searching 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 was used to design a vehicle suspension system in this project. Design objectives such as bouncing acceleration and bouncing transmissibility are introduced for accessing comfortability of the suspension. While the searching space of the parameters is very large, the solution space is very tight. Therefore, the objective function is slightly reformed so that small variation in design paparemters Ks and Cs can reflect considerably large difference in fitness. And the added restrictions on suspension working space and natural frequency increase the complexity of the problem. In all simulation runs, it can be observed that the genetic algorithms has been able to find optimal suspension systems which are similar to those found with local optimization search methods. These results are encouraging and suggest that GA can be easily used in other complex and realistic designs often encountered in the engineering.

As we all know that GA has its drawback: it requires large computing effort. However, the efficiency of the GAs can be improved by monitoring previously analyzed designs so as to avoid re-computing the fitness for an existing optimal design. To further improve efficiency and consistency in the results, the GA parameters, such as population size and mutation probability, may be tuned more effectively. Most importantly the GA results show that there is potential to incorporate global optimization methods for suspension system design.

7 FUTURE WORKS

Refine the vehicle dynamics model to include more detailed suspension characteristics;

Expand the vehicle dynamics model to pitch plane;

Introduce more objectives regarding ride comfort such as minimizing jerk, RMS of seat acceleration;

Expand the parameter ranges;

Convert the Matlab code calculating vehicle dynamics to Fortran such that more advanced GA toolbox such as NSGA II can be used.

8 REFERENCES

Houck, C.R., Joines, J.A, and Kay M.G, (1995), *A genetic algorithm for function optimization: a Matlab implementation*, NCSU-IE TR 95-09, North Carolina State University.

Joines, J. and Houck, C. (1994) . *On the use of non-stationary penalty functions to solve constrained optimization problems with genetic algorithms*. IEEE International Symposium Evolutionary Computation, Orlando, FL, pp. 579-584.

Various authors, *Handbook of Evolutionary Computation,* The Institute of Physics Publishing, 1998-2000

Cebon, D (1999) *Handbook of Vehicle-Road Interaction*

Wong, J.Y (2001) *Theory of Ground Vehicles* 3rd edition

Baumal, A.E and McPhee, J.J *Application of Genetic Algorithms to the Optimization of an Active Vehicle Suspension Design*