DETC2003/DAC-48714

ROBUST OPTIMIZATION OF AN AUTOMOBILE VALVETRAIN USING A MULTIOBJECTIVE GENETIC ALGORITHM

Emre Kazancioglu, **Guangquan Wu**, **Jeonghan Ko**, **Stanislav Bohac**,
Zoran Filipi, **S. Jack Hu**, **Dennis Assanis** and **Kazuhiro Saitou***
Department of Mechanical Engineering
University of Michigan, Ann Arbor, MI 48109-2125, USA
{ckazanci,kazu}@umich.edu

ABSTRACT

A robust optimization of an automobile valvetrain is presented where the variation of engine performances due to the component dimensional variations is minimized subject to the constraints on mean engine performances. The dimensional variations of valvetrain components are statistically characterized based on the measurements of the actual components. Monte Carlo simulation is used on a neural network model built from an integrated high fidelity valvetrain-engine model, to obtain the mean and standard deviation of horsepower, torque and fuel consumption. Assuming the component production cost is inversely proportional to the coefficient of variation of its dimensions, a multi-objective optimization problem minimizing the variation in engine performances and the total production cost of components is solved by a multi-objective genetic algorithm (MOGA). The comparisons using the newly developed Pareto front quality index (PFQI) indicate that MOGA generates the Pareto fronts of substantially higher quality, than SQP with varying weights on the objectives. The current design of the valvetrain is compared with two alternative designs on the obtained Pareto front, which suggested potential improvements.

INTRODUCTION

An automobile valvetrain (Figure 1) is a high-speed cam-follower mechanism responsible for synchronizing the intake and exhaust of gases in cylinders of an internal combustion engine. The dimensional variations of the valvetrain components cause undesired deviations of engine performance from the design specifications. While tightening dimensional tolerances of each component decreases the performance variations, it also increases the manufacturing cost of the component. It is therefore desirable to understand the trade-offs between the performance variation and the component

manufacturing cost, thereby tightening the tolerances of *only* the components that have a large influence on the performance variations, while maintaining the desired level of mean performances.

The conventional robust optimization approaches formulate this type of problem as a minimization (or maximization) of an objective function combining measures of mean performances, performance variations, and manufacturing cots, typically as a weighted sum of these measures [1-4]. A trade-off curve (Pareto front) of these measures can be generated by the multiple optimization runs with different weight values. However, the generation of a Pareto front with sufficiently large number of points can be very inefficient, since one point in a Pareto front requires an optimization run with certain weight values. Further, the obtained points can be too unevenly spread to understand the trade-offs since the weight can only indirectly control the spread of points. Finally, the method severely suffers when the Pareto front is concave viewing from the utopia point [5,6,7].

This paper aims at demonstrating the effectiveness of a multi-objective genetic algorithm (MOGA) [8-15] in generating Pareto fronts for a robust optimization problem of an automotive valvetrain, where the variation in engine performances and the production cost of components are minimized subject to the constraints on mean engine performances. Based on the measured statistics of dimensional variations in valvetrain components, Monte Carlo simulation is used on a neural network model of a high fidelity valvetrain-engine model, in order to obtain the mean and standard deviation of three measures of engine performances: horsepower, torque and fuel consumption.

Using the newly developed Pareto front quality index (PFQI) that quantifies the closeness to the utopia point and overall spread of the points in a Pareto front, the results by

---

* corresponding author

MOGA are compared to the Pareto fronts generated by a sequential quadratic programming (SQP) with varying weights on the objectives. The comparisons indicate that MOGA generates the Pareto fronts of substantially higher quality than SQP, with the same number of function evaluations. According to the obtained Pareto front, the original design of the valvetrain is found to be sub-optimal and different scenarios for improvement is discussed.
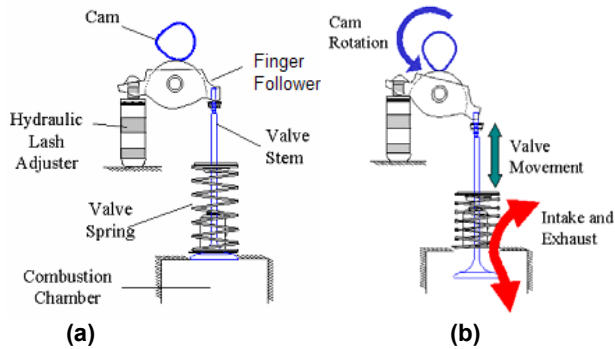
Figure 1. Typical automotive valvetrain. (a) components and (b) motion.

RELATED WORK

Robust Optimization in Engineering Design

The robust optimization problem is usually solved by using an aggregate objective function to capture both maximum performance and minimum performance variation due to variations in the design variables. The calculation of the performance variation typically requires direct sampling of variables with Monte Carlo simulation or the approximation of the performance function and the distribution of the design variables.

Gu *et al.* [1] presented a response-surface based safety optimization and robustness process using regression and Latin hyper cube sampling methods. They employed a Monte Carlo simulation based stochastic simulation method along with a sequential quadratic programming approach to solve the reliability based design optimization model for robust system parameter design. Lee and Park [2] formulated a single, multi-objective function to incorporate the mean and standard deviation of the original objective function to optimize a structural problem in a robust manner using recursive quadratic programming. Sunderasan *et al.* [3] and Su *et al.* [4] developed robust optimization methods where the objective function is a convex combination of the original objective functions and their variations. A weighting factor determines the trade-off between the robust optimum and the optimum where robustness is not considered. Messac and Yahaya [16] formulated the robust design optimization problem from a multi-objective perspective using a Physical Programming approach.

While the above work treats both mean and standard deviation of the performances as objectives, the trade-off between performance variation and component tolerances would be of more interest when an existing baseline design is to be improved for robustness. Considering such scenarios, this paper formulates the problem as minimization of performance variation and manufacturing cost (as a function of component tolerances) subject to the constraints on shift of the mean performance from the baseline values.

Multiobjective Genetic Algorithm

Zitler and Lother [8] compared different multi-objective evolutionary algorithms (MOEA) where the extended 0/1 knapsack problem is taken as a basis and introduces Strength Pareto Evolutionary Algorithm (SPEA). Shim *et al.* [9] introduced Pareto-based continuous evolutionary algorithms for multi-objective optimization problems having continuous search space. Tan *et al.* [10] developed a GUI based multiobjective evolutionary algorithm toolbox which is freely available. Horn *et al.* [11] introduced a nitched Pareto genetic algorithm to tackle multi-objective optimization problems incorporating Pareto domination in the selection operation and nitching to maintain diversity. An extensive discussion on MOGAs and suggestions on customized forms of MOGAs for a variety of applications can be found in [12]. One such algorithm, non-dominated sorting genetic algorithm (NSGA) [13], is used in this paper.

Due to the stochastic nature of the algorithm, Pareto fronts generated by MOGA can be different in each run with varying qualities. However, very few researchers have proposed metrics to measure the quality of the Pareto fronts. Wu and Azarm [15] introduced one such metric called the inferiority index, which requires rather complicated calculation. The Pareto front quality index (PFQI) proposed in this paper quantifies the quality of a set of points in a Pareto front in terms of their closeness to the utopia point and overall spread in a simple intuitive manner, and will be used to compare different sets of Pareto points.

HIGH-FIDELITY VALVETRAIN-ENGINE MODEL

### Simulation Models of Valvetrain-Engine Dynamics

This study is conducted on the valvetrain system of a Ford Duratec 2.5L V6 SI engine, released in 1994. The engine has a maximum power output of 125 kW at 6250 rpm and 220 Nm of torque at 4250 rpm and is used in the Mercury Mystique, Ford Contour and Ford European Mondeo. Its specifications are listed in Table 1, and the photos of its valvetrain assembly and camshaft are shown in Figure 2.

The main function of the valvetrain system is to control the flow of intake and exhaust gases by opening and closing the valves, which is obtained by transforming the rotational camshaft motion into linear motion of the valve. A valvetrain is a complex and nonlinear dynamic system due to the existence of lash, part deformation by dynamic forces, and uncertain

hydraulic pressure. In order to accurately predict the valve motion, commercial software GT-Vtrain [17] is used to a build a lumped-parameter dynamic model of the Ford Duratec valvetrain.   Figure 3 shows a schematic of the valvetrain model, and Figure 4 shows the required input parameters, some of which will be regarded as design variables during optimization.

**Table 1. Specifications of Ford Duratec engine.**

|	
Displacement [cc]	2544
Bore [mm]	82.4
Stroke [mm]	79.5
Compression ratio	9.7:1
Cylinder Arrangement	V6
Angle btw two banks [of cyl. deg.]	60
Firing orders	1-4-2-5-3-6
Valve gear type	Chain driven, dual overhead camshaft
Number of Valves	24
Fuel injection	Multi-point sequential
Rated power/Speed [kW/rpm]	125/6250
Max torque/Speed [Nm/rpm]	220/4250

| (a) | (b) |
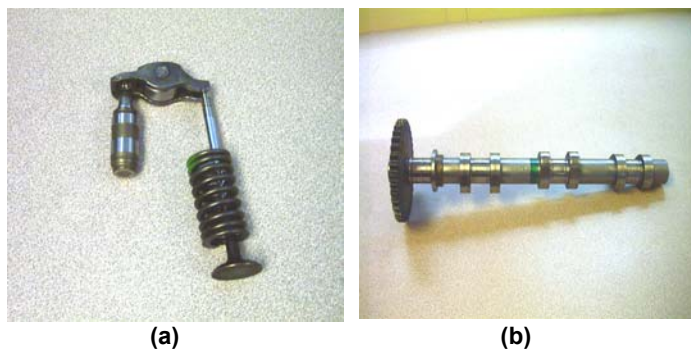
Figure 2. photos of valetrain components of Ford Duratec engine. (a) valvetrain assembly and (b) cam shaft.
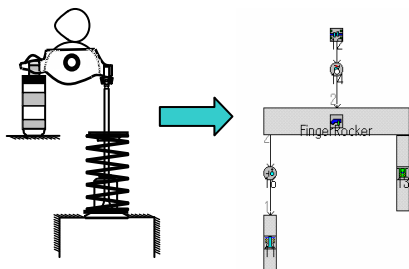
Figure 3. Lumped parameter dynamic model of a valvetrain by GT-Vtrain [18].
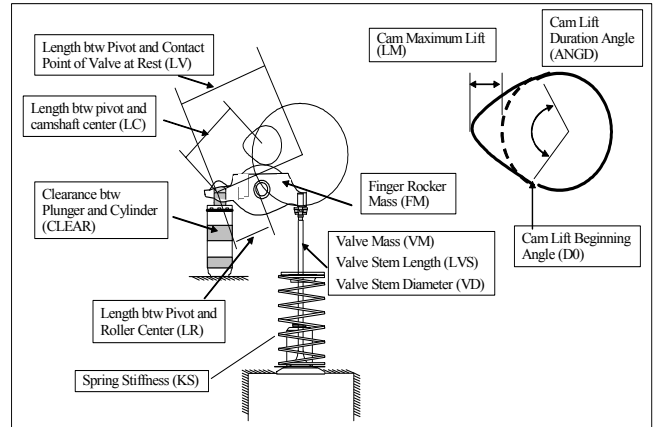
Figure 4. Input parameters of the valvetrain model.
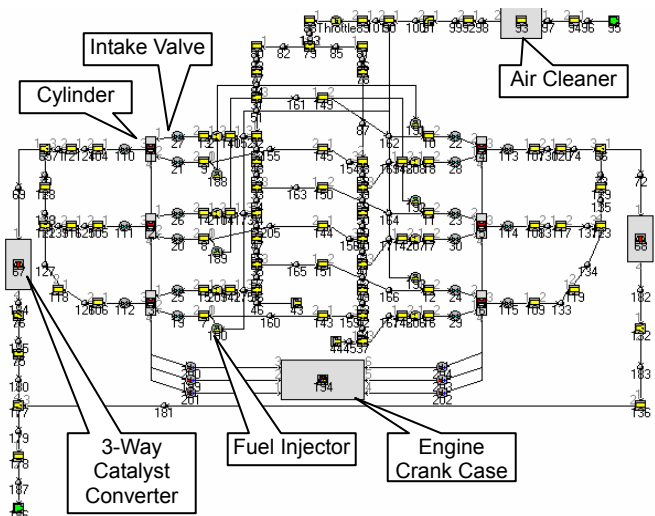
Figure 5. Engine model constructed in GT-Power [18].

The engine system simulation is constructed using GT-Power software tool [17]. The high fidelity simulation is based on one-dimensional gas dynamics to represent the flow in the piping, and the thermodynamic in-cylinder model augmented by sub-models for specific engine phenomena such as combustion, heat transfer and emissions.   The major components of the engine system are air cleaner, throttle body, intake and exhaust manifolds, cylinders, valves, injectors and catalyst converters, which are illustrated in Figure 5. The main outputs of the engine model relevant for this study are horsepower, torque and fuel consumption.   The model constants were calibrated based on experimental data obtained from the engine set-up at the W. E. Lay Automotive Laboratory, University of Michigan.   Figure 6 illustrates the agreement between the simulated and measured pressure in the cylinder after calibration.   The ability of the simulation to accurately predict the overall engine system behavior, including its sensitivity to variations of valve timings, was confirmed

through comparisons against a comprehensive set of measured engine process parameters.

The valvetrain model is integrated to the engine model in order to simulate the effect of dimensional variations of the valvetrain components on the engine performances. The Ford Duratec engine has a variable intake system in which one of the intake ports, called the secondary port, is only effective at engine speeds higher than 3500 rpm. The cam profiles generated at the end of the valvetrain simulation are fed into the engine model. Figure 7 illustrates the integrated high fidelity valvetrain-engine model.
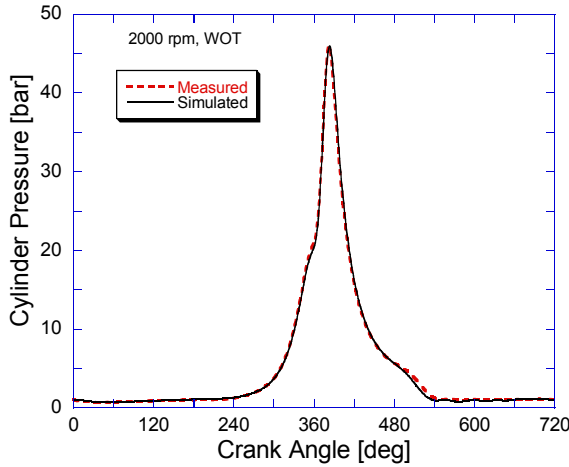

Figure 6. Comparison of measured and predicted cylinder pressure.
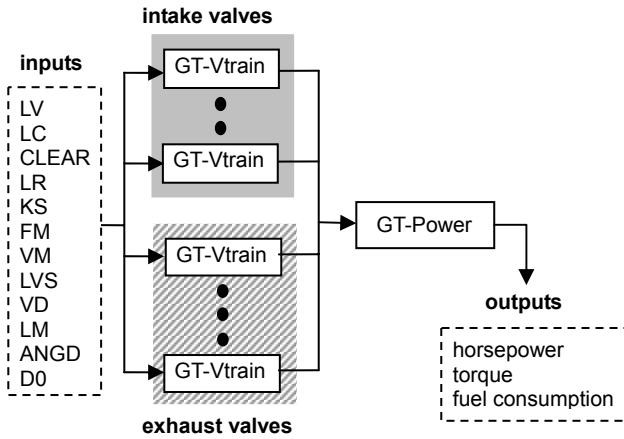

Figure 7. High fidelity valvetrain-engine model.

Statistical Model of Dimensional Variations

To statistically characterize the dimensional variations of the input parameters in Figure 4, a number of intake and exhaust valvetrain components are measured using a coordinate measurement machine (CMM). The measurement data are collected for the following subset of the input parameters due to their high sensitivity to the engine performances:

- Valve mass (VM) [g]
- Valve stem length (LVS) [mm]
- Valve stem diameter (VD) [mm]
- Cam maximum lift multiplier (LM)
- Cam lift duration angle (ANGD) [deg]
- Cam lift beginning angle (D0) [deg]

As shown in Figure 8, the value of VM is strongly dependent on the values of LVS and VD (Figure 8 (a)(b)), and the value of *LM* is weakly dependent on the value of ANGD (Figure 8 (c)). The linear regression provides the following relationships:

$$\mu_{VM} = -38.504836 + k_1(LVS)(VD)^2 + k_2 LVS + k_3(VD)^2 \quad (1)$$

$$\sigma_{VM} = 0.076739769 \quad (2)$$

$$VM = N(\mu_{VM}, \sigma_{VM}) \quad (3)$$

where $k_1 = 7.3 \times 10^{-5}$, $k_2 = 6.7 \times 10^{-5}$, $k_3 = 0.2 \times 10^{-5}$. Similarly for LM, the multiple linear regression model provides:

$$\mu_{LM} = 0.8483 + 0.3035 \, ANGD \quad (4)$$

$$\sigma_{LM} = 0.004117422 \quad (5)$$

$$LM = N(\mu_{LM}, \sigma_{LM}) \quad (6)$$

Equations 2 and 5 show the standard error terms which are an estimate of the standard deviation of least squares point estimate. The rest of the dimensions are regarded as independent.

**Table 2. Statistics of measurement data of independent parameters.**

| parameter	mean (μ)	std. dev. (σ)
LVS [mm]	115	0.03
VD [mm]	5.9	0.01
ANGD [deg]	0.5	0.003
D0 [deg]	0	2

Neural Network Model of Engine Simulation Model

Since the model in Figure 7 requires 18 valvetrain simulation runs followed by an engine simulation run to obtain one output, the calculation of performance variation using the model would be computationally very expensive. Accordingly, a surrogate model of the simulation is built using a feed forward neural network trained by one step secant backpropagation [18,19], available in MATLAB Neural Network Toolbox.
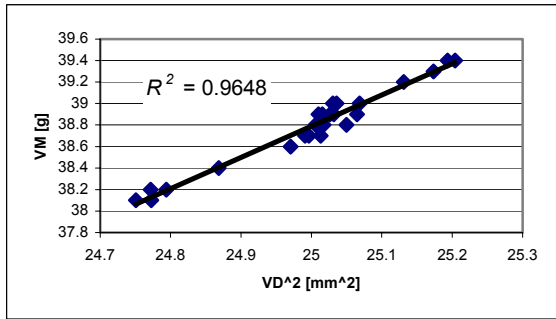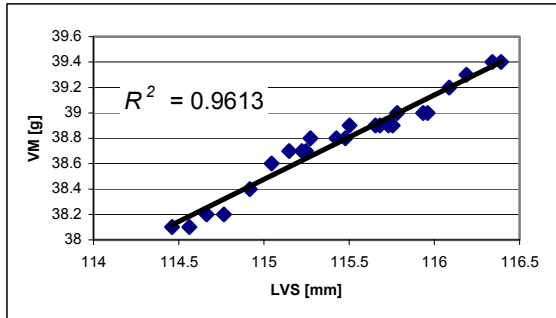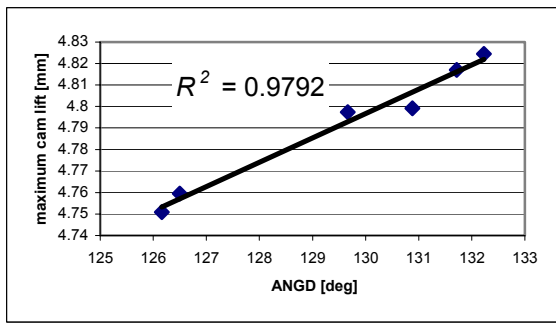
Figure 8. Correlation of measured data. (a) valve mass (VM) and valve stem diameter (LVS), (b) valve mass (VM) and valve stem length (LVS), and (c) maximum cam lift and cam lift duration angle (ANGD). R in each figure is the residual.
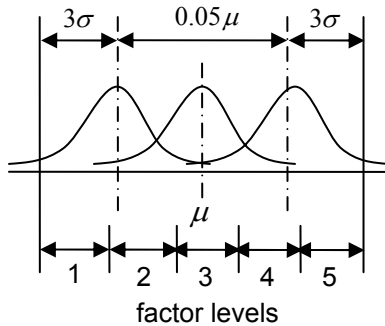

Figure 9. Range of independent parameters and factor levels for sampling training data of neural network model.

The training data (pairs of the inputs and outputs of the simulation model) is sampled by using the five-level full factorial design of the four independent parameters in Table 2. Training of the neural network and subsequent optimization was performed at the engine operating condition of 3000 rpm, full load. As illustrated in Figure 9, each parameter is assumed to vary within the range $[0.975\mu\text{-}3\sigma, 1.025\mu\text{+}3\sigma]$, and each factor level is defined as the range equally dividing this into five. At each factor level, a value is uniformly sampled in the corresponding range. The values of the dependent input parameters are calculated using the regression models (Equations (1) and (2)). The five-level full factorial design with the four parameters gives the total of 625 training data, each consisting of 72 inputs (4 parameters for 18 valves) and 3 outputs (horsepower, torque, fuel consumption). Two sets of such 625 data (independently sampled) are produced using the high fidelity simulation and subsequently used to train the neural network. Figure 10 shows the horsepower output for 50 random unseen inputs, using the trained neural network model and the simulation model. The outputs of the neural network model match very well with the corresponding ones from the simulation model.
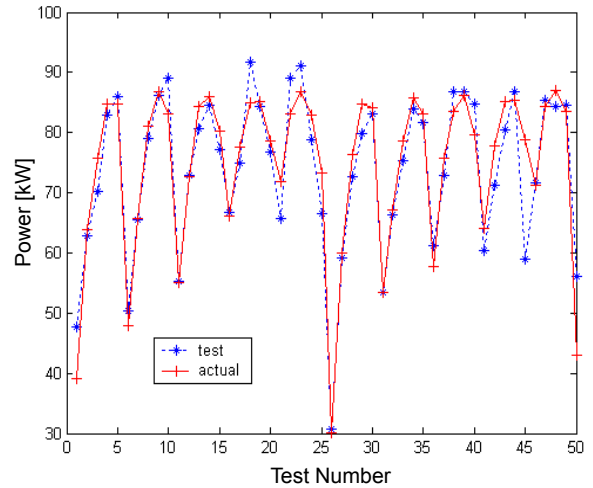

Figure 10. Horsepower output for 50 random inputs, using neural network model (dotted line) and original simulation model (solid line).

ROBUST OPTIMIZATION OF VALVETERAIN SYSTEM

Design Variables

The mean and standard deviation of the four independent parameters in Table 2, valve stem length (LVS), valve stem diameter (VD), cam lift duration angle (ANGD), and cam lift beginning angle (D0) are selected as design variables for robust optimization of the V6 engine valvetrain system. Table 3 shows the upper and lower bounds of the design variables (corresponding to both intake and exhaust valvetrain components) considered in the following examples. The ranges are set such that sampled values of the input parameters by

Monte Carlo simulation do not fall outside the ranges for which the neural network model was trained.

Table 3. Lower and upper bounds of design variables.

Design var.	Lower Bound	Upper Bound
μ_{LVS}	113.5	117
μ_{VD}	5.78	6.02
μ_{ANGD}	0.46	0.55
μ_{D0}	-7	7
σ_{LVS}	0.005	0.1
σ_{VD}	0.001	0.05
σ_{ANGD}	0.0005	0.01
σ_{D0}	0.1	2

Objective Functions and Constraints

Considering a scenario where an existing baseline design is to be improved for robustness, the problem is formulated as minimization of performance variations and manufacturing costs subject to the constraints on the mean performance measures. The variations of the engine performance measures (f_1) are computed by Monte Carlo simulation applied to the neural network model:

$$f_1 = \sigma_{HP} + \sigma_T + \sigma_{FC} \qquad (7)$$

where subscripts HP, T, FC indicate horsepower, torque, and fuel consumption, respectively. The manufacturing cost f_2 is calculated by summing the inverses of the coefficients of variation (σ/μ) of LVS, VD, ANGD and D0 (simply $1/\sigma$ for D0, since mean of D0 is zero) [20]:

$$f_2 = \frac{\mu_{LVS}}{\sigma_{LVS}} + \frac{\mu_{VD}}{\sigma_{VD}} + \frac{\mu_{ANGD}}{\sigma_{ANGD}} + \frac{1}{\sigma_{D0}} \qquad (8)$$

Finally, the constraints on the mean performance measures are given as:

$$\mu_{HP} \geq 70 \text{ [kW]}$$
$$\mu_T \geq 160 \text{ [Nm]} \qquad (9)$$
$$\mu_{FC} \leq 285 \text{ [g/(kW-Hr)]}$$

In the following examples, these constraints are formulated as the third objective function f_3 by summing the corresponding penalty terms:

$$f_3 = \max\{0, 70 - \mu_{HP}\} + \max\{0, 160 - \mu_T\} + \max\{0, \mu_{FC} - 285\} \qquad (10)$$

Note that $f_3 = 0$ if all constraints are satisfied.

Multiobjective Genetic Algorithm (MOGA)

Multiobjective genetic algorithms (MOGA) are an extension of genetic algorithms that do not require multiple objectives to be aggregated to one value, for example, as a weighted sum. Instead of static aggregates such as a weighted sum, MOGA dynamically determine an aggregate of multiple objective values of a solution based on its relative quality in the current population, typically as the degree to which the solution dominates[1] others in the current population. The following examples use non-dominated sorting genetic algorithm (NDSGA) [6], where the quality of a solution is measured in terms of the number of solutions dominating it in the current population, as outlined below:

1. Create a population P of n chromosomes (an encoded representation of design variables) and evaluate their values of objective functions.
2. Rank each chromosome c in P according to the number of other chromosomes dominating c (rank 0 is Pareto optimal in P). Store the chromosomes with rank 0 into set O. Also, create an empty subpopulation Q.
3. Select two chromosomes c_i and c_j in P with probability proportional to n-$rank(c_i)$ and n-$rank(c_i)$.
4. Crossover c_i and c_j to generate two new chromosomes c_i' and c_j' with a certain high probability.
5. Mutate c_i' and c_j' with a certain low probability.
6. Evaluate the objective function values of c_i' and c_j' and store them Q. If Q contains less than m new chromosomes, go to 3.
7. Let $P \leftarrow P \cup Q$ and empty Q, Rank each chromosome in P and remove m chromosomes with lowest ranks from P.
8. Update set O and increment the generation counter. If the generation counter has reached a pre-specified number, terminate the process and return O. Otherwise go to 3.

Since the design variables in Table 3 are continuous variables, *real-coded* chromosomes are used - a chromosome is simply a vector of design variables in Table 3. Accordingly, the two crossover methods known as effective for real-coded chromosomes, heuristic crossover and quadratic crossover, are adopted in the step 4 above. In the following, $\boldsymbol{f} = (f_1, f_2, .., f_n)$ denotes a vector-valued objective function to be minimized and r is a uniform random number in [0,1].

Heuristic crossover creates n children from two parent vectors, where each child is an "extension" of the better parent in the descend direction of the corresponding objective function. For each objective function f_i, two parent vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ create a child vector $\boldsymbol{y_i}$ by randomly picking a point along the descend direction:

---

[1] For a vector-valued function $\boldsymbol{f} = (f_1, f_2, .., f_n)$ to be minimized, a point $\boldsymbol{x}$ dominates $\boldsymbol{y}$ if $f_i(\boldsymbol{x}) < f_i(\boldsymbol{y})$ for all $i = 1, 2, \ldots, n$.

$$y_i = \begin{cases} r(x_1 - x_2) + x_1 & \text{if } f_i(x_1) < f_i(x_2) \\ r(x_2 - x_1) + x_2 & \text{otherwise} \end{cases} \quad (11)$$

Quadratic crossover [21] creates n children from three parents, where each child is the optimizer of the quadratic approximation of the corresponding objective function. For each objective function f_i, three parent vectors x_1, x_2 and x_3 can create a child vector y_i with the j-th component y_{ij} as:

$$y_{ij} = \begin{cases} -\dfrac{b_{ij}}{2a_{ij}} & \text{if } a_{ij} > 0 \\ z_{ij} & \text{otherwise} \end{cases} \quad (12)$$

where z_{ij} is *some* projection of x_{1j}, x_{2j}, or x_{3j} in the direction of the decreasing objective function, and a_{ij} and b_{ij} are the coefficients of the quadratic and linear terms in the quadratic approximation of f_i near x_{1j}, x_{2j}, and x_{3j}:

$$a_{ij} = \frac{1}{x_{3j} - x_{2j}} \left\{ \frac{f_i(x_{3j}) - f_i(x_{1j})}{x_{3j} - x_{1j}} - \frac{f_i(x_{2j}) - f_i(x_{1j})}{x_{2j} - x_{1j}} \right\}$$
$$b_{ij} = \frac{f_i(x_{2j}) - f_i(x_{1j})}{x_{2j} - x_{1j}} - a_{ij}(x_{2j} + x_{1j}) \quad (13)$$

Further details of quadratic crossover are given in [21].

Mutation in the step 5 is done as a uniform mutation:

$$y_j = lb + (ub - lb)r \quad (14)$$

where lb and ub are the lower and upper bounds of the the j-th design variable. If a variable y_j after crossover and mutation is out of bounds, it is simply rounded to the nearest boundary value to maintain the feasibility.

PARETO FRONT QUALITY INDEX (PFQI)

Definition

The set O returned at the termination of MOGA in the previous subsection contains the non-dominated solutions that the algorithm had encountered before termination. Due to the stochastic nature of the algorithm, however, the contents of O are likely different in each run and so are the qualities of resulting Pareto fronts. To quantitatively assess the performance of MOGA runs, the Pareto front quality index (PFQI) is introduced. PFQI quantifies the quality of points in the Pareto front in terms of 1) their closeness to the utopia point and 2) the range and 3) the evenness of the spread, thus providing effective global assessment of the Pareto front.

Let f_i^{min} and f_i^{max} be the lower and upper bounds of the i-th component of a vector-valued objective function $f = (f_1, f_2, ... f_n)$

to be minimized, and PF be a set of N points in a Pareto front normalized with f_i^{min} and f_i^{max}:

$$PF = \left\{ \hat{f}^j \mid \hat{f}^j = \frac{f^j - f^{min}}{f^{max} - f^{min}}; j = 1, 2, ... N \right\} \quad (15)$$

where $f^{min} = (f_1^{min}, ..., f_n^{min})$ and $f^{max} = (f_1^{max}, ..., f_n^{max})$. Let us define the n *extreme points* in the objective function space as:

$$f^{ext_i} = (f_1^{min}, ..., f_{i-1}^{min}, f_i^{max}, f_{i+1}^{min}, ..., f_n^{min}); i = 1, ... n \quad (16)$$

Note that the utopia point f^{min} and the i-th extreme point f^{ext_i} becomes the origin $(0, ..., 0)$ and the i-th unit vector $(0, .., 0, 1, 0, ..., 0)$ in the normalized objective function space, respectively. Then, the closeness of the points in PF to the (normalized) utopia point can be given as the average distance to the origin:

$$d_C = \sum_{j=1}^{N} \frac{\left\| \hat{f}^j \right\|}{N} \quad (17)$$

and the range of the points in PF can be measured as the sum of the distances between the (normalized) i-th extreme point and the point in PF that is closest to it.

$$d_R = \sum_{i=1}^{n} \frac{\min\limits_{j \in \{1, .., N\}} \left\| \hat{f}^j - \hat{f}^{ext_i} \right\|}{n} \quad (18)$$

Similarly, the evenness of the spread of the points in PF can be measured as the maximum distance to between two neighboring points in PF:

$$d_E = \max_{j \in \{1, .., N\}} \left(\min_{k \in \{1, .., N\}, k \neq j} \left\| \hat{f}^k - \hat{f}^j \right\| \right) \quad (19)$$

Finally, the Pareto front quality index (PFQI) is simply defined as the sum of d_C, d_R, and d_E:

$$PFQI = d_C + d_R + d_E \quad (20)$$

Note that, according to the formulation small PFQI values indicate better quality Pareto fronts.

Example

To illustrate the calculation of PFQI according to Equations (15)-(20), let us consider a simple Pareto front shown in Figure 11, consisting of points A, B and C in a two-dimensional normalized objective function space. The coordinates of points A, B, and C are (0.25, 0.75), (0.25, 0.50),

and (0.75,0.25), respectively. As described earlier, the origin O is the utopia point, and two unit vectors (1,0) and (0,1) correspond to the extreme points of the first objective (denoted as E_1) and the second objective (denoted as E_2). From Equation (17) we have:

$$d_C = \frac{1}{3}(OA+OB+OC) \approx 0.7134 \tag{21}$$

Since C is the closest to the extreme point E_1 and A is the closest to the extreme point E_2, Equation (18) gives:

$$d_R = \frac{1}{2}(E_1C+E_2A) \approx 0.3536 \tag{22}$$

From Figure 11, B is the closest to A, A is the closest to B, B is the closest to C, hence:

$$d_E = \max(AB,AB,BC) = BC \approx 0.5590 \tag{23}$$

Therefore, the PFQI value for the Pareto front in Figure 11 is given as:

$$PFQI = d_C + d_R + d_E \approx 0.7134 + 0.3536 + 0.5590 = 1.626 \tag{24}$$
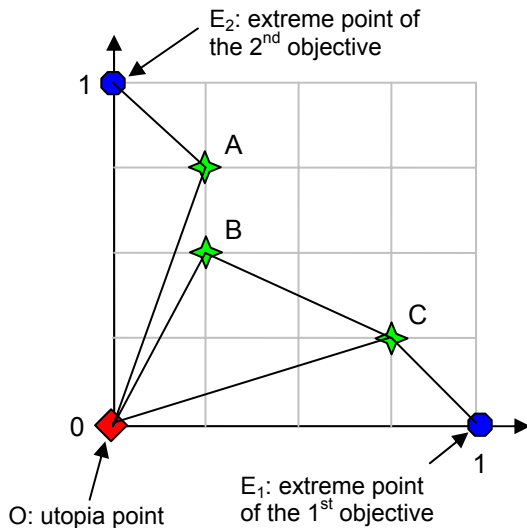

Figure 11. Example Pareto front within normalized space of two objective functions, consisting of 3 points A, B, and C.

RESULTS

Pareto Front Generation by MOGA

The results of the robust optimization of the valvetrain system are provided in this section. The Pareto fronts generated by MOGA are compared to the ones by SQP with varying weights on the objectives, using the Pareto front quality index developed above. The following results are obtained with the software implemented on MATLAB.

For MOGA, three types of tests runs are performed with a population size of 100 and the maximum numbers of generations being 20, 40 and 60. Each test is repeated until the total number of function evaluations becomes approximately 2500, 5000 and 7500, and the mean and standard deviation of the resulting PFQI are calculated. Table 4 lists the values of MOGA parameters.

For SQP, a weighted sum of two objective functions f_1 (Equation 7) and f_2 (Equation 8) is used as the single objective function, with five different values of weights in Table 5, where the values assume f_1 and f_2 are normalized to vary within [0,1]. For each set of weights w_1 and w_2, SQP runs are repeated 20 times to obtain 100 solutions, from which non-dominated points are extracted to calculate one PFQI value. Since SQP never successfully terminates by satisfying the optimality criteria, the maximum number of function evaluations is imposed to force the termination of the algorithm. Accordingly, three PFQI values are obtained by the results with 2500, 5000 and 7500 function evaluations.

Table 4. MOGA parameters for test runs.

Mutation probability	0.01
Crossover probability	0.9
Population size	100
Max. # of generations	20, 40, and 60

Table 5. Weights for two normalized objectives f_1 and f_2 for SQP runs.

#	1	2	3	4	5
w_1	0.7	0.6	0.5	0.4	0.3
w_2	0.3	0.4	0.5	0.6	0.7

Tables 6-8 show the statistics of PFQI by MOGA with the heuristic crossover (GAHX) and the quadratic crossover (GAQX), and the PFQI value by SQP, with 2500, 5000, and 7500 function evaluations, respectively. An example of Pareto front for each number of function evaluations are shown in Figures 12-14. Three observations can be made from these results:

1. **Validity of the definition of PFQI:** The visible improvements in the shape of Pareto fronts by GAHX and GAQX over 2500, 5000, and 7500 function evaluations are successfully captured as the corresponding decrease in the mean values of PFQI. Conversely, the visible deterioration of the shape of Pareto fronts by SQP is also well captured as the increase in the PFQI values.

2. **Superiority of GA over SQP in Pareto front generation**: For the same number of function evaluations, GA with

either crossover methods clearly generates higher quality Pareto front than SQP, particularly in terms of the number and the spread of the points in the Pareto front. Between two crossover methods, the quadratic crossover performed slightly better than the heuristic crossover in terms of both mean and standard deviation of PFQI values.

3. **Effect of the number of function evaluations on Pareto front quality:** Since MOGA never loses the good points encountered during the iteration, its performance tends to improve with more function evaluations as indicated both in the mean and standard deviations of PFQI values in Tables 6-8. On the other hand, SQP may move to a worse point with more function evaluations, for example, as indicated by the deterioration of Pareto front by SQP in Figure 14 as compared to the one in Figure 12. Consequently, the PFQI for the SQP runs with 7500 function evaluations is higher than the one obtained with 2500 function evaluations. However, a caching mechanism similar to MOGA could be implemented between multiple SQP runs to remedy this situation.

Table 6. PFQI values with 2500 function evaluations. GAHX and GAQX stand for MOGA with the heuristic crossover the quadratic crossover, respectively.
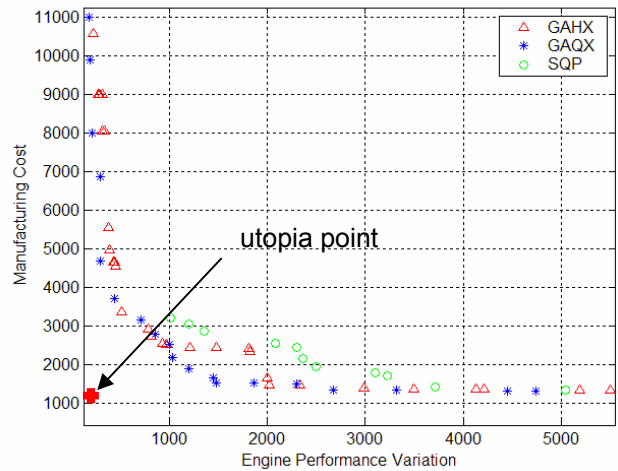
	μ	σ
GAHX	0.73074	0.15232
GAQX	0.66368	0.11843
SQP	0.95126	

Table 7. PFQI with 5000 function evaluations.

	μ	σ
GAHX	0.70804	0.12788
GAQX	0.57512	0.11078
SQP	1.0047	

Table 8. PFQI with 7500 function evaluations.

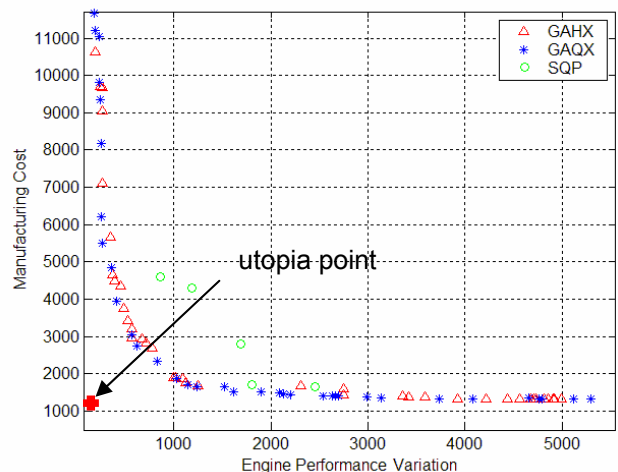|	μ	σ
GAHX	0.59464	0.12035
GAQX	0.50054	0.10949
SQP	1.2399	

Figure 12. Pareto fronts with 2500 function evaluations. GAHX and GAQX stand for MOGA with the heuristic crossover the quadratic crossover, respectively.
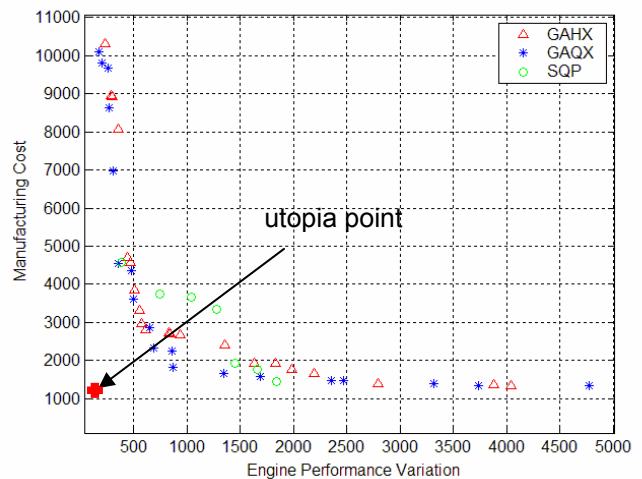
Figure 13. Pareto fronts with 5000 function evaluations.

Figure 14. Pareto fronts with 7500 function evaluations.

9

Examination of Pareto Optimal Designs

Figures 12-14 show the Pareto fronts of the performance variation and manufacturing cost is a simple convex curve. The high curvatures near the utopia point in the Pareto fronts indicate the following basic trend:

- Reducing the performance variations from high to medium values can be done with almost no increase in the manufacturing cost.
- Reducing the performance variations from medium to small values can cause a significant increase in the manufacturing cost.

Figure 15 shows the point (852.59, 3630.9) corresponding to the current valvetrain design in Table 2 computed by the high-fidelity valvetrain-engine model, and two potential alternative designs (Alt-1 and Alt-2), relative to the best Pareto front by GAQX with 7500 function evaluations. The location of the current design suggests that there is room for improvements with respect to the definition of the present objective functions. By better allocating the tolerances and shifting the means of the design variables, for instance, the same level of the engine performance variations can be attained at a lower manufacturing cost (Alt-1). Alternatively, at the same manufacturing cost, it is possible to attain less variations of the engine performance (Alt-2). The values of the design variables for these alternative designs are shown in Table 9. Figure 15 shows the outputs of the alternative designs obtained by the neural network model (NN) and by the high fidelity valvetrain-engine model (VE). The proximity of corresponding points obtained by the NN and the high-fidelity VE model demonstrates the validity of the approach based on the use of the surrogate model for generating Pareto fronts.

Table 9. Values of design variable for alternative designs.

Design Var.	Current	Alt-1	Alt-2
μ_{LVS}	113.5	117	117
μ_{VD}	5.78	5.78	5.78
μ_{ANGD}	0.46	0.55	0.55
μ_{D0}	0	-7	-7
σ_{LVS}	0.03	0.09339	0.066
σ_{VD}	0.01	0.00644	0.0051
σ_{ANGD}	0.003	0.00273	0.0005
σ_{D0}	2	0.1	0.1

While the two alternative designs are identical with respect to the mean values of the design variables, they allocate different tolerances to achieve the respective engine performances. In both cases, the tolerances for VD, ANGD and D0 are tightened, whereas the LVS tolerance is loosened.
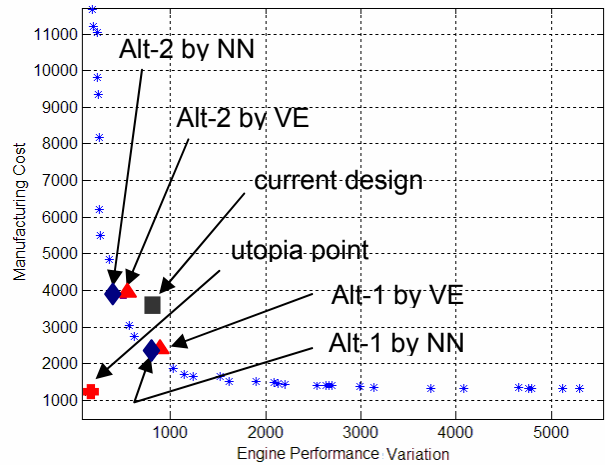
Figure 15. Performances of the current valvetrain design and two possible alternative designs (Alt-1 and Alt-2) obtained by the neural network model (NN) and by the high fidelity valvetrain-engine model (VE).

From the viewpoint of valvetrain mechanism, the loosening of the valve stem length (LVS) could be interpreted as the effect of hydraulic lash adjustors to compensate for these variations. The tightening of the tolerances for the stem diameter required by Alt-1 or Alt-2 is mild. In contrast, the large tightening in cam lift duration angle (ANGD) and cam lift beginning angle (D0) tolerances indicate that their effects on reducing the variations in engine performances offset the resulting increases in the manufacturing cost. The fact that the mean D0 value converges to a lower bound shows that for given conditions advanced phasing of cams leads to better performance.

Table 10 lists measures of engine performance (*i.e.*, power, torque and fuel consumption) for the alternative designs, obtained by the neural network and the high fidelity valvetrain-engine models. The results in the table indicate good agreement between the neural network model and the valvetrain-engine model that was previously validated against experimental measurements. The alternative designs are in fact superior to the current design with respect to all engine performances in both mean and standard deviation, although the improvements in mean performances are minute since they are regarded as constraints rather than objectives.

Table 10. Engine performances of current design and alternative designs.

	Power (kW)		Torque (N-m)		Brake Specific Fuel Consumpt.	
	μ	σ	μ	σ	μ	σ
Current	77.41	4.343	185.38	2.128	278.9	4.139
Alt-1 by NN	79.40	3.345	190.20	1.549	271.64	3.428
Alt-1 by VE	78.41	3.527	190.22	1.689	271.65	3.893
Alt-2 by NN	78.91	1.557	190.29	0.806	270.76	1.737
Alt-2 by VE	77.56	1.789	189.42	0.993	274.93	1.934

While the two design alternatives on the obtained Pareto front clearly suggested potential design improvements, a caution must be taken in interpreting the results in Table 9. Since Equation (8) simply sums the inverses of coefficients of variation of all design variables, the resulting value does not account for the relative difficulties of manufacturing processes that affects variables. For example, tightening the tolerance of VD by turning could be far easier than tightening the tolerances of ANGD by cam surface grinding or D0 by shrink-fit assembly. Although weights can be given to each term in Equation (8), obtaining reasonable weight values may not be straightforward, since the manufacturing cost of a part depends on many factors such as the type of processes and the volume of production.

SUMMARY

The multi-objective optimization framework offers a means to compare design alternatives by providing trade-offs between multiple objectives. In this paper the multi-objective robust optimization of an automotive valvetrain is presented using multi-objective genetic algorithm (MOGA). To objectively access the quality of Pareto fronts generated by MOGA, the Pareto front quality index (PFQI) is developed that quantifies the closeness to the utopia point and the range and evenness of the spread.

The results confirmed the validity of the PFQI for capturing the overall quality of a Pareto front, and also revealed substantial benefits of using multi-objective Gas, both in terms of the quality of the generated Pareto fronts and the required computational resources. On the other hand, SQP with varying weights failed to capture the while Pareto front since the even spread of weights do not usually correspond to the even spread of points in the Pareto front [5,6,7]. Further, the stochastic nature of the objective functions and constraints severely hindered the effectiveness of the gradient-based approach resulting in sub-optimal points.

From an engine design viewpoint, the obtained Pareto front captured a trade-off between the cost of manufacturing and the variations of the engine performances. This allows the designer to efficiently examine multiple Pareto optimal designs among which a desired design can be selected by considering the maximum allowable engine performance variation and/or manufacturing cost. The results indicates the variation of the engine performance can be reduced by tightening tolerances of the cam profile, both its beginning angle and the duration, and by relaxing the tolerance of the valve stem length, which can offset the increase in the manufacturing cost due to tighter cam tolerances.

ACKNOWLEDGMENTS

REFERENCES

[1] Gu, L., Yang, R. J., Tho, C. H., Makowski, M., Faruque, O., and Li, Y., 2001, "Optimization and Robustness for Crashworthiness of Side Impact," *International Journal of Vehicle Design,* Vol. 26, No. 4, pp. 348-360.

[2] Lee, K. and Park, G., 2001, "Robust Optimization Considering Tolerances of Design Variables," *Computers and Structures,* No. 79, pp. 77-86.

[3] Sunderasan, S., Ishii, K., and Houser, D. R., 1993, "A Robust Optimization Procedure with Variation on Design Variables and Constraints," *Advances in Design Automation,* Vol. 65, No. 1, pp. 379-386.

[4] Su, J., and Renaud, J. E., "Automatic Differentiation in Robust Optimization," *AAIA,* Vol. 35, pp. 1072-1079.

[5] Yu, P., 1985, *Multiple-Criteria Decision Making*, Plenum Press, New York.

[6] Athan, T. and Papalambros, P., 1996, "A Note on Weighted Criteria Methods for Compromise Solutions in Multi-Objective Optimization," *Engineering Optimization*, Vol. 27, pp. 155-176.

[7] Das, I., 1997, "Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems," *Structural Optimization*, Vol. 14, No. 1, pp. 63-69.

[8] Zitzler, E. and Lother, T., 1999, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257-271.

[9] Shim, M., Suh, M., Furukawa, T., Yagawa, G., and Yoshimura, S., 2002, "Pareto-Based Continuous Evolutionary Algorithms for Multiobjective Optimization," *Engineering Computations*, Vol. 19, No. 1, pp. 22-48.

[10] Tan, K. C., Lee, Tong, T. H., Khoo, D., and Khor, E. F., 2001, "A Multiobjective Evolutionary Algorithm Toolbox for Computer-aided Multiobjective Optimization," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Cybernetics, Vol. 31, No. 4.

[11] Horn, J., Nafpliotis, N., and Goldberg, D. E., 1994, "Niched Pareto Genetic Algorithm for Multiobjective Optimization," *Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence,* Vol. 1, pp. 82-87.

[12] Coello, C. A. C., van Veldhuizen, D. A., and Lamont, G. B., 2002, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic/Plenum Publishers, New York.

[13] Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T., 2000, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," *Proceedings of the Parallel Problem Solving from Nature VI Conference*, Paris, France, pp. 849-858.

[14] Kuparti, A., Azarm, S., and Wu, J., 2002, "Constraint Handling Improvements for Multiobjective Genetic Algorithms," *Structural and Multidisciplinary Optimization*, Vol. 23, pp. 204-213.

[15] Wu, J., and Azarm, S., 2001, "On a New Constraint Handling Technique for Multi-Objective Genetic Algorithms," *Proceedings of the Genetic and Evolutionary Computation Conference,* pp. 741-748.

[16] Messac, A. and Yahaya, A. I., 2002, "Multiobjective Robust Design Using Physical Programming," *Structural and Multidisciplinary Optimization,* Vol. 23, No. 5.

[17] Gamma Technologies, Inc. www.gtisoft.com/index.html.

[18] Battiti, R., 1992, "First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method," *Neural Computation*, Vol. 4, No. 2, pp. 141–166.

[19] Villiers, J. and Barnard, E., 1992, "Backpropagation Neural Nets with One and Two Layers," *IEE Transactions on Neural Networks,* Vol. 4, No. 1.

[20] Chase, K. W. and Greenwood, W. H., 1975, "Design Issues in Mechanical Tolerance Analysis," *Manufacturing Review*, Vol. 1, pp. 50-59.

[21] Adewuya, A. A., 1996, *New Methods in Genetic Search with Real-Valued Chromosomes*, Master's Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, June.