JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 3, June 2006

Fast Multidisciplinary Design Optimization via Taguchi Methods and Soft Computing

Paul P. Lin*
Cleveland State University, 2121 Euclid Ave., SH 232, Cleveland, OH 44115-2214

and

Kenol Jules†
NASA Johnson Space Center, Mail Code 0Z-4, 2101 NASA Parkway, Houston, TX 77058-3696

It is difficult to perform multidisciplinary design optimization using traditional search-based optimization techniques due to possible conflicts among objectives from different disciplines, the time consuming search, and the possibility of divergence. To overcome the difficulties, this paper presents simulation-based design optimization techniques using Taguchi methods and soft computing (i.e. fuzzy logic and neural networks). An aircraft engine cycle design optimization with four conflicting design objectives is used to validate the presented approach. The result shows significant performance improvement in optimizing single and multiple design objectives.

I. Introduction

THE emerging field of Multidisciplinary Design Optimization (MDO) seeks to improve design methodology to rapidly and efficiently explore multiple-dimension design spaces with the goal of increasing system performance significantly, thereby reducing end-product cost substantially. Search-based and simulation-based are the two major system design approaches. The former is traditional and mathematical, and has existed for a long time. The optimum solution has to do with the selected starting point, and the optimization method used. A possibility of divergence in solution seeking is a major drawback in this approach. In contrast, the simulation-based approach uses the analysis and evaluation of a candidate solution, and the assessment of the degree to which the candidate satisfies the requirement. This optimum design tool uses the simulation-based approach.[1] With this new approach, the optimum solution can be obtained in real time.

The traditional search-based optimization is a typical example of hard computing. In hard computing, the prime desiderata are precision, certainty and vigor. In contrast, in soft computing the principal notion is that precision and certainty carry a cost; and that computation, reasoning, and decision-making should exploit (whenever possible) the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth for obtaining low cost solutions.

Fuzzy logic and neural networks, the two major soft computing techniques, have very contrasting application requirements. Fuzzy systems are appropriate if sufficient expert knowledge about the process is available, while neural systems are useful if sufficient data are available or measurable. Furthermore, neural networks possess the ability to learn the input-output relationship. A trained neural network provides instantly input-to-output mapping with reasonably good accuracy, but without knowledge representation. Fuzzy logic, on the other hand, possesses the ability for knowledge representation and inference, but has no capabilities for automated learning. Thus, fuzzy logic and neural networks compensate each other in terms of information processing.

Received 2 February 2005; revision received and accepted for publication 7 April 2006.
* Professor and Chairman, Mechanical Engineering Department.
† Lead Increment Scientist.

A fast multidisciplinary system optimization tool is presented, and an aircraft engine cycle design optimization is used to validate the presented design approach. In addition, the two important issues in design optimization: global optimization versus local optimization and single-objective versus multiple-objectives are investigated.

II. The Design Optimization Methodology

In a typical multidisciplinary system, there are many disciplinary application modules, and several design objectives under each module. The goal is to develop a design tool that can perform fast-optimized multidisciplinary system design in real time within good accuracy.

The design tool is structured in a way that every module is optimized and trained separately so that if one module for a specific application is changed, the others can remain unchanged. Each application module has several design objectives. For instance, the engine cycle module has four design objectives: maximizing thrust, minimizing fuel consumption, minimizing emission, and minimizing jet velocity.

When dealing with multiple objectives, the optimum solution has to be compromised to resolve the potential conflict among the objectives. Two primary issues in multiple objective decision-makings are to acquire meaningful information regarding the satisfaction of the objectives by the various choices (alternatives) and to rank or weight the relative importance of each of the objectives. Bass and Kwakernaak[2] proposed the concept of rating and ranking of multiple-aspect alternatives using fuzzy sets. Later Yager[3] proposed a decision-making methodology based on fuzzy sets, which requires the ordinal information about the ranking of preferences and importance weights. In 1993, Sakawa[4] extended the Yager's work by making multiple-objective optimization interactive using fuzzy sets.

Design alternatives are usually evaluated at the preliminary design stage using primarily fuzzy sets. However, at the design optimization stage, there are no design alternatives to evaluate because the design configuration is already determined. In this case, fuzzy sets will not be as useful. But, fuzzy logic when combined with other techniques can play an important role in design optimization.

The presented simulation-based optimization is accomplished by combining Taguchi methods, fuzzy logic and neural networks. They are briefly described below:

A. Taguchi Methods

Taguchi methods are a statistical process that perturbs a parameter in order to study its influence on the overall output. Taguchi methods' strength lies in their ability to extract relatively large amount of information from limited experiments (fractional factorial compared to full factorial). The basic tools used to obtain the information are orthogonal arrays and linear graphs. Three valuable pieces of information are generated from a Taguchi analysis:

1. Which factors or parameters are significant to the output (or objective functions).
2. The relative significance of those factors.
3. Which direction for levels of those factors will lead to further improvement or optimization to the design.

An orthogonal array contains the number of experimental runs, the number of levels of each input factor or parameter (such as two levels: high and low), and the number of columns in the array.[5] In an orthogonal array, every input factor is placed in one of the columns. A linear graph contains the relationship of input factor interactions. Taguchi has created a transformation of the repetition data to another value, which is a measure of the variation present. This transformation is the signal-to-noise (S/N) ratio.[5–6] By examining the S/N ratios, the significant factors can be identified.

For each design objective, the Taguchi methods identify the degree of significance of each parameter to its output. This parameter significance is then normalized to form a so-called *parameter significance index*, which contains the following three trends/categories: the higher the better (HB), the closer to the mean the better (MB), and the lower the better (LB). Thus, for each input parameter the significance index indicates the trend and the relative degree of significance. The procedure for performing Taguchi analysis is briefly described below with an illustrative example.

Step 1: Find the appropriate Taguchi orthogonal array

For an illustration purpose, Table 1 shows a popular orthogonal array, L_{27} (3^{13}) in which the subscript "27" stands for 27 experiment runs, "3" stands for three levels (1 for low, 2 for medium and 3 for high), and the superscript "13" stands for 13 columns which are for control factors and their interactions.

Table 1 Taguchi's orthogonal array L_{27} (3^{13}).

Run	Column no.												
no.	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2	3
11	2	1	2	3	2	3	1	2	3	1	2	3	1
12	2	1	2	3	3	1	2	3	1	2	3	1	2
13	2	2	3	1	1	2	3	2	3	1	3	1	2
14	2	2	3	1	2	3	1	3	1	2	1	2	3
15	2	2	3	1	3	1	2	1	2	3	2	3	1
16	2	3	1	2	1	2	3	3	1	2	2	3	1
17	2	3	1	2	2	3	1	1	2	3	3	1	2
18	2	3	1	2	3	1	2	2	3	1	1	2	3
19	3	1	3	2	1	3	2	1	3	2	1	3	2
20	3	1	3	2	2	1	3	2	1	3	2	1	3
21	3	1	3	2	3	2	1	3	2	1	3	2	1
22	3	2	1	3	1	3	2	2	1	3	3	2	1
23	3	2	1	3	2	1	3	3	2	1	1	3	2
24	3	2	1	3	3	2	1	1	3	2	2	1	3
25	3	3	2	1	1	3	2	3	2	1	2	1	3
26	3	3	2	1	2	1	3	1	3	2	3	2	1
27	3	3	2	1	3	2	1	2	1	3	1	3	2

Step 2: Construct the orthogonal array

Starting with the first parameter (column 1) in Table 1, the values for the first nine experiments or cases are set at the low end (i.e. the lowest in the specified range), whereas the values for the next nine are set at the middle (i.e. the mean), and the remaining other nine are set at the high end (i.e. the highest). The process is repeated for each of the other parameters.

Step 3: Find the corresponding output

A program is written to run the 27 cases, and the corresponding output for each case is recorded. Thus, for each design optimization objective, there are a total of 27 outputs to be evaluated.

Step 4: Perform Taguchi analysis

When performing Taguchi analysis, the S/N (Signal-to-Noise) ratio for each case is calculated. The S/N ratio is an indication of significance. There are three different types of S/N ratios: the-larger-the-better, the-nominal-the-better and the-smaller-the-better, depending on the type of objective.[6] For instance, the formula for calculating the S/N ratio for the-smaller-the-better type is:

$$\eta = -10 \log_{10} \left[\left(\sum_{i=1}^{n} y_i^2 \right) / n \right] \tag{1}$$

where n is the number of experiment runs for each level (i.e. 1, 2 or 3), and y_i is the corresponding output value for each run. Thus, the S/N ratios for the first parameter at low end will be the η value obtained from runs 1 through 9, whereas the ratios at the medium will be the η value obtained from runs 10 through 18, and those at the high end will be the η value obtained from runs 20 through 27. Likewise, the S/N ratios for the second parameter at the low end will be the η value obtained from the following nine different runs: 1, 2, 3, 10, 11, 12, 19, 20, and 21. The process is repeated for each parameter at each level. Afterward, the η values for each parameter are compared at the three different levels. If, for instance, the η value at the low end is the highest, then this parameter is said to be *the lower the better* in optimizing the output (design objective) value.

In addition to calculating the S/N ratios for each parameter and identifying the trend for each parameter (i.e. the lower the better, the closer to the mean the better or the higher the better), the Taguchi analysis also generates parameter significance from which significant parameters can be identified. In other words, the Taguchi analysis identifies which parameters are more significant than others, and uses their trends (low, medium or high) to help optimize a design objective. In MDO where conflicting design objectives often exist, parameter settings become complicated, which is better to be handled by fuzzy logic. The parameter significance will be further explained in the section of global optimization versus local optimization.

B. Fuzzy Logic

Fuzzy logic is a mathematical technique for understanding, and controlling specific manipulation of continuously variable truth-values.[7] In more specific terms, fuzzy logic is all about the relative importance of precision. Thus, fuzzy logic measures the truth of a given situation as a matter of degree. Between the input and the output, there is a black box that does the work through the use of if-then rules, which embody the knowledge that governs the action of the system that is being described or modeled. For this work, fuzzy logic is used to set each design parameter at optimum, and also to resolve the conflict among several design objectives. The input for the fuzzy logic contains membership functions of each input variable, and the output also contains membership functions of each output variable. The degree of membership is between -1 and 1 for each input variable, and 0 and 1 for each output variable. The Taguchi analysis generates significance index value, which falls in one of three categories: the higher the better (HB), the mean the better (MB), and the lower the better (LB). In HB and LB, the value is positive and negative, respectively. However, the value in MB can be a small positive or negative number meaning that the parameter value should be set slightly above or below the mean in the specified parameter range. Therefore, the fuzzy inference system for this work was designed to have three input memberships (HB, MB and LB), and three output memberships (High, Medium and Low). The input is the normalized significance index value (from Taguchi analysis) between -1 and 1, where -1 stands for the extreme value in LB, and 1 stands for the extreme value in HB. The output represents the optimum value for each parameter, which is normalized between 0 and 1. The output value of 1 indicates that the parameter value should be set at highest within the specified range, whereas the output value of 0 means that the parameter value should be set at the lowest within the specified range.

For single-objective optimization, each parameter has one fuzzy logic input variable (the significance index value) and one output variable. However, for multiple-objectives optimization, say four (4), for example, there are four input variables (the significance indices for objectives 1 to 4) and one output variable. In this particular case (four objectives), the fuzzy inference system takes four values into consideration at the same time, and yields a compromised solution as output. The procedure for applying fuzzy logic is briefly described below.

Step 1: Define input and output membership functions

A membership function essentially embodies all fuzziness for a particular fuzzy set; its description is the essence of a fuzzy property or operation. Therefore, it is important to adequately define the membership functions. Some typical membership functions are such as stepwise, triangular, trapezoidal, sigmoid curve, and Gaussian curve. Type of membership function, such as triangular or Gaussian curve, is not critical in designing a fuzzy inference system. However, having appropriate number of membership functions for input and output is important. Each type of membership function is specified by several parameter values. For instance, a triangular membership function uses three parameter values (i.e. three vertex locations) to define its shape. These three values were first assumed and then tested by observing the output values. This process usually takes several modifications until all the membership functions are properly defined.

Step 2: Develop the fuzzy rules

The if-then fuzzy rules are derived from the knowledge obtained via Taguchi analysis. For a single-objective case, they can be as simple as the following three rules:

Rule 1: If (input1 is HB) then (output1 is High)
Rule 2: If (input1 is MB) then (output1 is Medium)
Rule 3: If (input1 is LB) then (output1 is Low)

For design optimization of single objective, there will be one input (parameter significance) and one output (parameter adjustment), and the three rules can be stated more specifically as follows:

Rule 1: If (parameter significance is HB) then (parameter adjustment is High)
Rule 2: If (parameter significance is MB) then (parameter adjustment is Medium)
Rule 3: If (parameter significance is LB) then (parameter adjustment is Low)

Likewise, for design optimization of four objectives, there will be four inputs (one parameter significance for each objective), and one output (parameter adjustment), and a total of 12 rules (three rules for each objective). In this case, the four inputs are taken into account at the same time to determine the best parameter adjustment for all objectives. In other words, the fuzzy logic resolves the conflicts among design objectives, and helps find the compromised solution.

Step 3: Run the fuzzy inference system

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions are made. There are two types of fuzzy inference systems: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined. The former, chosen for this work, is most commonly seen fuzzy methodology. The fuzzy inference system performs the following tasks:[8] (1) fuzzify the inputs, (2) apply fuzzy operation, (3) apply implication method, (4) apply aggregation method, and (5) defuzzify the output. The end result is a crisp value for each parameter.

In this work, the fuzzy inference system first fuzzifies the normalized parameter significance index values between -1 and 1, and finally defuzzifies the parameter adjustment value between 0 and 1. This parameter adjustment value is then used for optimum parameter settings later. The main concept in the simulation-based approach is that more significant factors should be adjusted more, and the amount of adjustment should be proportional to their degree of significance. Fuzzy logic is the best candidate to deal with degree of membership (i.e. the degree of significance, in this case).

The inter-parameter constraints are handled by the use of parameter significance and fuzzy logic. For instance, to optimize objective 1, parameters 1 and 2 may be completely in conflict with each other. Thus, if parameter 1 needs to be set as high as possible, then parameter 2 will have to be set as low as possible. In this case, parameter 1 will have a positive parameter significance value, while parameter 2 will have a negative value. The next task is parameter settings, which should be according to their respective degree of significance. In other words, if parameter 1 is the most significant parameter in the system, then it will be set at the very high end. Parameter 2, on the other hand, will be set low, but not at the very low end because this parameter is not most significant. Note that the fuzzy inference system uses the degree of parameter significance as input, and generates a crisp output value for parameter adjustment.

C. Neural Networks

Artificial neural networks are created to mimic the neural system in human brains. The network consists of many interconnected nodes, similar to neurons in the human brain. Each node assigns a value (known as weight) to the input from each of its counterparts. As these values (i.e. weights) are changed, the network can adjust the way it responds. A typical network usually has at least three layers: the input layer, the hidden layer, and the output layer.[6] The first layer is a vector containing input data, and the last layer is a vector containing the desired output. The number of neurons in the input layer equals the input variables. Likewise, the number of neurons in the output layer equals the number of output variables. The second layer is known as the hidden layer, which is essential to handle the non-linear relationship between inputs and outputs. When the relationship is highly non-linear, two or even three hidden layers may be used. It should be noted that some researchers do not consider input data as a layer. Thus, their typical network architecture becomes two layers (hidden layer and output layer). In terms of network training, backpropagation neural network[9] (BPNN) is the most widely used net, since it yields accurate outputs when the training data are rich. A trained neural network can be used to provide instant input-to-output mapping.

D. Interaction Among the Three Techniques

The framework of this multidisciplinary system optimization consists of three phases: Taguchi analysis, fuzzy inference and neural network mapping. Taguchi analysis is performed first, which generates the normalized parameter significance index values. These values are then passed on to the fuzzy inference system in which every input value is fuzzified and then defuzzified to yield a crisp value between 0 and 1. Finally, these crisp values are used for parameter setting in which 0 and 1 correspond to the lowest and highest values in the specified range, respectively. The trained neural network mapping instantly gives the output values of the optimum design (when a database is available). This instant mapping gives the user a chance to fine-tune his or her design without having to go through the optimization process one more time.

III. Global Optimization versus Local Optimization

The search-based approach often leads to a local optimum. A common searching technique in global optimization is known as multi-start, in which the user keeps changing the starting point until no better solution can be found. In contrast, the simulation-based approach automatically leads to a global optimum solution.

A. Parameter Significance

The Taguchi analysis generates the significance value for each design parameter at its low (L), mean (M) and high (H) levels. The so-called signal-to-noise (S/N) ratios for each parameter at these three levels are calculated. There are several ways to calculate the S/N ratio depending on the purpose of the design objective (i.e. minimizing, maximizing, averaging, etc.). The formulas for calculating S/N ratios are designed such that higher ratios always represent higher significance regardless of maximization or minimization.

The S/N ratios at the low end, middle and high end are first evaluated to determine the possible trend: the lower the better (LB), the closer to the mean the better (MB) and the higher the better (HB). However, in reality, the parameter whose S/N ratio is the highest can be anywhere between the low and high ends. To find where the highest S/N ratio is, the three S/N ratio values (at the low end, middle and high end) are fitted by a second order curve, as shown in Fig. 1, from which the location of the highest value can be easily determined.

For each parameter, the parameter significance is calculated by taking the difference between the S/N ratio of the current design baseline and the maximum S/N ratio. Thus, every parameter has its parameter significance value. These values are then normalized (by dividing all values by the maximum) to form the Parameter Significance Index Values (PSIV). The fitted S/N ratio curve remains the same for all cases, but the PSIV varies whenever the design baseline is different. Thus, a parameter that is most significant with the first design baseline, and could become most insignificant if the second design baseline happens to be very close to the optimum point.

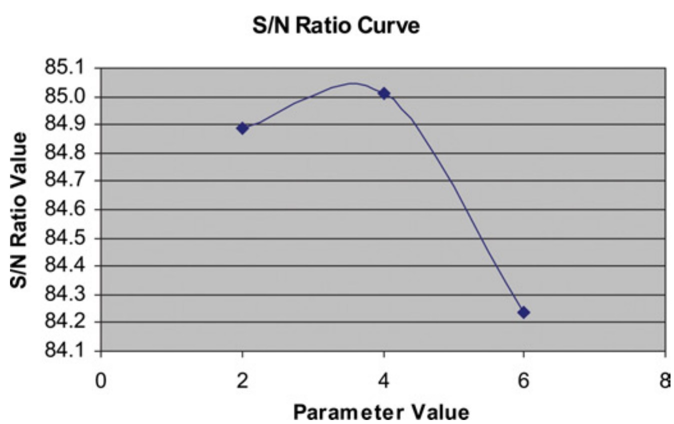

Fig. 1 The fitted S/N ratio curve.

B. Parameter Settings

The last and most important task in the simulation-based optimization is parameter settings. Note that the significance values for all parameters are normalized separately at the high, medium and low levels. They carry a positive or negative sign. A positive or negative parameter significance value signifies the necessity of having positive or negative parameter adjustment, respectively. The fuzzy inference system transforms from the normalized significance index value in the space of $[-1, 1]$ to parameter adjustment in the space of $[0, 1]$. In terms of parameter adjustment, value of "1" means that the parameter is best set at the point where the parameter significance value is at maximum, whereas value of "0" means that the parameter is best set at where it is now. The current design baseline is always compared with the optimum point. Thus, some parameters need to be increased from their current values, while others need to be decreased, instead. The amount of parameter adjustment depends on the parameter's degree of significance, which is gracefully handled by fuzzy logic. Since the parameter adjustment is continuous, the presented simulation-based approach is, indeed, for continuous optimization.

IV. Single-Objective Optimization versus Multiple-Objectives Optimization

In comparison with single-objective optimization, multiple-objectives optimization is much more complex. The complexity arises as the number of objectives increases. The more objectives the system has to optimize, greater the possibility of conflict will be. In addition, the user's preference weights on each objective also contribute to the complexity. Generally speaking, multiple-objectives optimization can be classified into the following two types:

A. Multiple-Objectives Optimization with No Preference

In this case, all design objectives are treated equally important. The fuzzy inference will resolve the conflicts among different objectives with equal weight. Thus, in the case of two objectives, there will be two input parameters fed into the inference system to produce a single crisp value as output.

B. Multiple-Objectives Optimization with Preferences

In the search-based multiple-objectives optimization, the user simply pre-multiplies the preference weights on each objective function to form the so-called pseudo objective function F

$$F = \sum w_i f_i \tag{2}$$

where w_i is the preference weight on the i^{th} objective, and f_i is the function of the i^{th} objective.

In the case of two objectives, for instance, 0.7 on objective 1 (i.e. more preferred) and 0.3 on objective 2 (less preferred), the pseudo objective function becomes:

$$F = w_1 f_1 + w_2 f_2 = 0.7 f_1 + 0.3 f_2 \tag{3}$$

The difficulty in optimizing the above new function is that the user has to know in advance, the approximate order of magnitude for each objective function. Otherwise, the weight distribution will not be as desired.

In contrast, the simulation-based multiple objective optimization deals with the preference weights at the end of the optimization process. For instance, if the fuzzy output value of objectives 1 and 2 are Z_1 and Z_2, respectively, and the preference weights on objectives 1 and 2 are w_1 and w_2, respectively, then the overall fuzzy output value Z can be obtained from the following weighted average formula:

$$Z = (w_1 Z_1 + w_2 Z_2)/(w_1 + w_2) \tag{4}$$

where Z_i is in the range of [0, 1]. Note that $w_1 + w_2 \neq 1$, but $[w_1/(w_1 + w_2)] + [w_2/(w_1 + w_2)] = 1$

In the case of simultaneously optimizing two extremely conflicting objectives with no preference, parameter A might have to be set at the high end (i.e. $Z_1 = 1$) to optimize objective 1, and also set at the low end (i.e. $Z_2 = 0$) to optimize objective 2. Substituting $Z_1 = 1$ and $Z_2 = 0$, and $w_1 = 1$ and $w_2 = 1$ (i.e. equal weight) into Eq. (4) gives the value of 0.5 for Z, which means that this parameter should be set in the middle of the range as a compromised solution. Now if preference weights are changed, such as $w_1 = 2$, and $w_2 = 1$, then the overall output value Z will become 0.667, instead of 0.5. This indeed, reflects the shift on the user's preference.

It should be noted that Eq. (4) is essentially equivalent to taking the linear approximation in a multi-variable space. The fuzzy logic's membership functions can be linear, but they are generally non-linear. Therefore, in the case of no preference, it is not recommended to use Eq. (4), by making all w_{i} equal. In the case of preference, one can edit he fuzzy logic rules and place different weights on each rule. But, this will require the user to know how to edit the fuzzy rules during the course of optimization, which could be cumbersome for some users. In this case, Eq. (4) can be used as an alternative. The Eq. (2), often used in the search-based multiple-objectives optimization, should not be used in the simulation-based multiple-objectives optimization because the preference weights placed on each objective have no bearing on determining the parameter significance, which means that the user's preference will not be taken into account.

In the case of having preference, Eq. (4) is a preferred alternative due to the fact that the user can obtain the desired optimum solution simply by placing different preference weights on each objective without having to perform the optimization all over again. Furthermore, the equation guarantees that the overall fuzzy output value Z to be always bounded between 0 and 1.

V. Illustrative Example

As a result of Taguchi analysis, parameter significance values are calculated and normalized between −1 and 1. An example containing these values and their trends are listed Table 2.

Note that LB, MB and HB stand for the higher the better, the closer to the mean the better and the higher the better, respectively. If the trend for a parameter is LB, then its significance index value will be negative. Likewise, if the trend is HB, then its index value will be positive. The index values that are slightly above or below 0 will fall into the category of MB.

In this example, there are 14 design parameters and 4 design objectives. In terms of optimizing a single objective, for instance, objective 1, par_2 (i.e. parameter 2) is LB, and has an index value of −1, which means that it should be set at the very low end. Also, par_6 is MB, and has the index value of 0.020, which means that it should be set slightly above the midpoint. In terms of optimizing two objectives such as objectives 1 and 2 with equal preference weight, par_5 should be set at the very high end for optimizing objective 2 alone, but it should also be set low for optimizing objective 1 alone. Thus, a conflict exists between these two objectives. In this case, fuzzy logic can gracefully resolve the conflict, and generate a compromised solution that is a little above the midpoint. If the user wishes to place more preference on objective 1, then the new compromised solution might be below the midpoint. As the preference weight ratio between the two objectives increases, the ability of the fuzzy inference system to resolve the conflict decreases. This is simply because the purpose of multiple-objectives optimization is to simultaneously

Table 2 Parameter significance index values.

	Objective 1		Objective 2		Objective 3		Objective 4	
	Trend	Index	Trend	Index	Trend	Index	Trend	Index
par_1	HB	0.273	LB	−0.221	LB	−0.571	LB	−0.305
par_2	LB	−1	MB	−0.088	HB	0.749	MB	−0.026
par_3	HB	0.237	HB	0.292	LB	−0.516	LB	−0.164
par_4	MB	0.093	MB	0.077	HB	0.229	MB	−0.044
par_5	LB	−0.592	HB	1	HB	0.191	HB	1
par_6	MB	0.020	HB	0.225	LB	−1	MB	0.028
par_7	LB	−0.178	MB	0.053	HB	0.110	MB	−0.075
par_8	MB	0.056	MB	0.034	LB	−0.248	MB	−0.063
par_9	HB	0.123	LB	−0.109	MB	0.087	MB	0.038
par_10	HB	0.142	LB	−0.490	LB	−0.383	MB	−0.060
par_11	LB	−0.134	MB	0.023	MB	0.071	LB	−0.129
par_12	HB	0.137	HB	0.253	HB	0.102	LB	−0.265
par_13	MB	−0.075	HB	0.110	HB	0.121	HB	0.058
par_14	HB	0.551	MB	−0.071	LB	−0.194	MB	−0.038

optimize or improve all design objectives. Greatly improving one objective at the big expense of the other is not a common practice in MDO.

VI. Results and Discussions

To evaluate this design tool, a design baseline was chosen. The baseline was run both at sea level and at an altitude of 1,067 m. This is because the designs are very different from each other. Therefore, the parameter significant index values must be generated separated for each case. Tables 3 through 6 show the optimization performance results for optimizing one, two, three and four objectives, respectively. The four conflicting design objectives are listed below:

Objective 1: To maximize the engine thrust
Objective 2: To minimize engine fuel consumption
Objective 3: To minimize the engine emission
Objective 4: To minimize the engine jet velocity

Design Baseline

The 14 parameters used as the design baseline for aircraft engine design are listed below:

par_1 = 0; 0.8, par_2 = 0; 1,067, par_1 = 3.75, par_2 = 0.8975, par_3 = 0.8, par_4 = 7, par_5 = 0.9075, par_6 = 0.9225, par_7 = 0.9125, par_8 = 3490, par_9 = 3245, par_10 = 2620, par_13 = 16, and par_14 = 3449 where par_1 and par_2 are Mach number and Altitude, respectively, which are evaluated at only two levels (i.e. Mach number = 1 at sea level and Mach number = 0.8 at altitude of 1,067 m). During the optimization process, par_3 through par_14 are free to move within their pre-specified ranges. These 12 parameters are such as low and high-pressure compressor ratios, low and high-pressure compressor efficiencies, bypass ratio, low and high-pressure turbine efficiencies, low and high-pressure turbine inlet temperatures, burner temperature, and inlet corrected airflow. For the sea level case, the initial four objective values are: thrust = 240,300 N., thrust specific fuel consumption = 0.8299, emission (NOX) = 0.16454 g nox/kg fuel, jet velocity = 72.94 m/sec. For the altitude of 1,067 m case, they are: 130,020 N., 1.202, 0.2543 g nox/kg fuel and 72.63 m/sec. Note that in this example, the allowable ranges for input variables are ±2% for all efficiencies, and ±10% for others.

Data Preparation

Two sets of aircraft engine data were used to train the input-output relationship. The first set about subsonic and above sea level (between 305 and 762 m in altitude) consists of 733 engine data (14 input parameters and 4 output parameters for each engine), and the second set about subsonic at sea level consists of 243 engine data. Some of the engine data already existed, but majority of them were newly created for this project only. To ensure even data distribution, the training data were originally based on a Taguchi orthogonal array, $L_{81}(3^{40})$, which contained 81 cases. The data were expanded later to include more combinations of parameter values. Running the aircraft engine

Table 3 Performance improvement in optimizing one design objective.

	Thrust (%)	Fuel (%)	Emission (%)	Jet vel. (%)
(a) Maximizing the thrust				
1,067 m	**12.57**	0.15	−19.52	−7.62
Sea level	**6.24**	−0.63	−10.09	−4.93
(b) Maximizing the fuel consumption				
1,067 m	0.95	**14.97**	−8.93	1.02
Sea level	−2.53	**9.38**	−4.84	3.55
(c) Maximizing the emission				
1,067 m	−6.40	3.44	**27.38**	4.34
Sea level	−4.44	0.73	**41.60**	4.02
(d) Maximizing the jet velocity				
1,067 m	−6.04	−4.72	13.24	**4.33**
Sea level	−9.01	2.27	25.80	**8.35**

Table 4 Performance improvement in optimizing two design objectives.

	Thrust (%)	Fuel (%)	Emission (%)	Jet vel. (%)
(a) Maximizing thrust, and minimizing fuel consuption				
1,067 m	**7.61**	**8.92**	−13.32	−4.51
Sea level	**1.20**	**6.48**	−4.59	0.09
(b) Maximizing thrust, and minimizing emission				
1,067 m	**2.63**	1.92	**5.82**	−2.05
Sea level	**−0.05**	2.34	**18.41**	0.23
(c) Maximizing thrust, and minimizing jet velocity				
1,067 m	**2.91**	−1.37	−2.16	**−2.28**
Sea level	**−2.86**	5.26	10.67	**2.94**
(d) Minimizing fuel consumption and emission				
1,067 m	−2.39	**9.53**	**10.76**	2.02
Sea level	−2.34	**5.68**	**21.43**	2.52
(e) Minimizing fuel consumption and jet velocity				
1,067 m	−2.58	**8.55**	4.40	**1.80**
Sea level	−5.23	**8.51**	14.20	**5.33**
(f) Minimizing emission and jet velocity				
1,067 m	−6.02	1.94	**19.37**	**4.05**
Sea level	−6.11	3.20	**31.70**	**5.66**

design program at NASA (without optimization) took about 15 to 20 minutes including data entry. For each engine, the 14 input parameter values and their corresponding four output values (i.e. thrust, fuel consumption, emission and jet velocity) were recorded in an Excel worksheet to later training.

The two sets of data were well trained using BPNN. They both passed the generalization test with RMS errors ranging between 1.2 and 3.7%. To best handle the non-linearity between the inputs and the outputs, two hidden layers, each with 20 neurons, were used in the BPNN. With rich data and use of two hidden layers, the training time was only about five to ten minutes for each data set. Good training essentially enables continuous input-output mapping.

In the following tables (Tables 3 to 6), a positive percentage of performance means that the objective has been optimized and the performance has been improved. This is regardless of maximization or minimization. For instance, a negative percentage for emission means that the emission was not improved, which results in increasing the emission. On the other hand, a positive percentage for emission means that the emission has been improved, which results in decreasing the emission. A design objective in bold face means that only that particular objective is optimized. Likewise, two objectives in bold face mean that the two are simultaneously optimized.

Table 5 Performance improvement in optimizing three design objectives.

	Thrust (%)	Fuel (%)	Emission (%)	Jet vel. (%)
(a) Maximizing the thrust, and minimizing the fuel consumption and emission				
1,067 m	**1.87**	**3.57**	**5.80**	−1.44
Sea level	**−0.19**	**3.53**	**16.91**	0.46
(b) Maximizing the thrust, and minimizing the fuel consumption and jet velocity				
1,067 m	**1.53**	**3.69**	0.41	**−1.29**
Sea level	**−2.57**	**5.75**	9.16	**2.73**
(c) Maximizing the thrust, and minimizing the emission and jet velocity				
1,067 m	**2.34**	1.85	**5.08**	**−1.89**
Sea level	**−2.38**	3.72	**17.29**	**2.33**
(d) Minimizing the fuel consumption, emission and jet velocity				
1,067 m	−2.30	**8.23**	**10.12**	**1.76**
Sea level	−3.96	**6.31**	**21.58**	**4.00**

Table 6 Performance improvement in optimizing four design objectives.

	Thrust (%)	Fuel (%)	Emission (%)	Jet vel. (%)
1,067 m	**1.28**	**3.48**	**5.43**	**−1.14**
Sea level	**−1.87**	**4.12**	**15.60**	**1.96**

As can be seen in the above four tables, the tool performs rather well in finding the optimum design, not only for single objective, but also for multiple objectives. The data shown in the tables are based on the assumption that all objectives are equally important. One can clearly discern the strong coupling between thrust and nozzle jet velocity; only one can be optimized at a time at the expense of the other one. This is simply because these two objectives conflict with each other. Jet velocity here is an indication of the engine noise level. It is worth noting that this design tool was able to find a solution, which decreases the fuel consumption and emission while increasing the thrust, with a small increase in jet velocity (about 1%) for the 1,067 m case (see Tables 5(a) and 6).

VII. Conclusions

The presented simulation-based approach combines Taguchi methods and soft computing techniques to perform multidisciplinary design optimization. More specifically,

1) Taguchi methods were used to generate the parameter significance index values
2) Based on the significance index values, fuzzy logic was used to set the design parameters at optimum
3) Fuzzy logic was also used to resolve the conflicts among different objectives
4) Neural networks were used to generate the instant input-output mapping. Such a capability allows the user to instantly evaluate the optimization performance, particularly in placing the preference weights and fine-tuning the optimum solution.

The presented simulation-based approach is superior to most traditional search-based optimization techniques in the following four aspects:

1) In seeking the optimum solution, this presented approach never diverges
2) The optimum solution is obtained in real time
3) The optimum solution is always global
4) The parameter significance index gives the user a good guidance to fine-tune the optimum solution.

The presented design approach intends to find the optimum solution in real time so as to reduce the design cost. Therefore, the solution thus obtained, without vigorous iterations, may not be at the very optimum mathematically. Nevertheless, the user can fine-tune the optimum solution or place different preference weights on each objective to achieve the desired optimum solution.

Acknowledgments

The authors would like to thank for the Propulsion Systems Analysis Office (PSAO) at NASA Glenn Research Center for its support of this project.

References

[1] Lin, P. P. and Jules, K., "A Simulation-Based Generic Tool for Multidisciplinary System Design," *8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000, AIAA 2000-4723.

[2] Bass, S., and Kwakernaak H., "Rating and Ranking of Multiple-aspect Alternatives Using Fuzzy Sets," *Automatica*, Vol. 13, 1977, pp. 47–58.

[3] Yager, R., "A New Methodology for Ordinal Multi-objective Decisions Based on Fuzzy Sets," *Decision Science*, Vol. 12, 1981, pp. 589–600.

[4] Sakawa, M., *Fuzzy Sets and Interactive Multiobjective Optimization*, Plenum Press, New York, 1993.

[5] Ross, P. J., *Taguchi Techniques for Quality Engineering*, 2nd ed., McGraw-Hill, 1996.

[6] Phadke, M. S., *Quality Engineering Using Robust Design*, Prentice-Hall Inc., 1989, pp. 108–111.

[7] Ross, T. J., *Fuzzy Logic with Engineering Applications*, McGraw-Hill, Inc., 1995.

[8] Jang, J. S., and Gulley, N., *Fuzzy Logic Toolbox User's Guide*, Mathworks Inc., 1997.

[9] Freeman J. A., and Skapura, D. M., *Neural networks: Algorithms, Application, and Programming Techniques*, Addison-Wesley Publisher, 1991.