Inverse Problems, Design and Optimization Symposium
Miami, Florida,U.S.A., April 16-18, 2007

MULTI-OBJECTIVE MDO SOLUTION STRATEGY FOR MULTIDISCIPLINARY DESIGN USING modeFRONTIER

Sumeet S. Parashar
Esteco North America
Livonia, MI, U.S.A.
sumeet.parashar@esteco.com

Nader Fateh
Esteco North America
Miami, Florida, U.S.A
nader.fateh@esteco.com

ABSTRACT

Design of complex engineering systems often involves multiple interacting disciplines and analyses. With the increase in computational power and sophisticated modeling techniques, the exploitation of the synergistic relationship that exists between coupled disciplines is becoming increasingly popular. While the interacting disciplines are trying to achieve one optimum system, the problem definition also often involves multiple conflicting objectives. The goal of this paper is to evaluate the application of a heuristic optimization strategy to solve tightly coupled Multidisciplinary Optimization (MDO) problems that involve multiple objectives. The multi-objective MDO problem is solved using the response surface methods and evolutionary algorithms available in modeFRONTIER [1]. A non-hierarchic tightly coupled MDO problem is formulated using Multidisciplinary Feasible (MDF) [2,3] formulation and solved using Non-dominated Sorting Genetic Algorithm (NSGA-II) [4]. NSGA-II allows the generation of evenly distributed Pareto designs in a faster and more efficient manner. The iterative convergence of coupling variables called System Analysis (SA) is reduced by utilizing response surface approximations, with subsequent update and refinement, thereby greatly increasing computational efficiency. The solution strategy is applied to a mathematical MDO test problem and the results are discussed.

INTRODUCTION

Today's sophisticated design process for complex engineering systems involves multiple analysis tools, engineering disciplines, and design teams. Design process is often characterized by decomposition of the system into smaller and more manageable subsystems. The decomposition cannot be considered completely without taking into account the complex interactions that exists between the decomposed subsystems. MDO [5] has emerged as an engineering discipline that focuses on development of new design and optimization strategies for such complex multidisciplinary systems. Depending on the type of subsystem interaction, systems could be decomposed in a hierarchic or non-hierarchic fashion. A non-hierarchic decomposition could result in one or more tightly coupled sub-system interactions. In a tightly coupled sub-system interaction, each sub-system analysis is dependent on the other through some coupling parameters or variables. Various formulations for posing and solving such tightly coupled MDO problems include Multidisciplinary Feasible (MDF) [2,3], Individual-Discipline Feasible (IDF) [2,3], All-At-Once (AAO) [2,3], Collaborative Optimization (CO) [6,7], and Concurrent Subspace Optimization (CSSO) [8-10]. In order to account for the coupling parameters, each of the above formulations either increases the problem size or requires an iterative convergence called SA at each design point. Either approach adds to the computational expense of the design process.

Most of the MDO formulations and strategies mentioned above were developed initially for a single system level objective function. However, while each subsystem is trying to contribute to a final optimum system, the problem definition often requires defining one of more objectives at subsystem or system level. For a multi-objective problem with conflicting objectives, there is not a single optimum but a set of optima called Pareto solutions. The Designer must choose one Pareto design as his final design based on his preferences. In situations where the designer is not sure of his preferences before the optimization process, he can generate the complete Pareto set of solutions and make an informed decision from the available Pareto designs. Generating the complete Pareto frontier for MDO problems can add further computational expense and may not even be possible under tight time and resource constraints.

In this paper a simple yet efficient solution technique is presented for effective generation of Pareto frontier for tightly coupled multi-objective MDO problems. The solution technique proposed here utilizes Non-dominated Sorting Genetic Algorithm (NSGA-II) for multi-objective optimization because of its ability to generate well distributed Pareto frontier without requiring any prior knowledge of the performance space. The solution technique is also designed to save time and computational expense by incorporating Neural Network approximations and hence avoid performing SA quite too often. The solution technique is implemented using the Multi-Objective Optimization and Design Environment (modeFRONTIER) software which allows the problem to be set up and solved very easily and efficiently. The details of the implementation are provided in the following sections. The method is applied to a moderate size coupled MDO test problem generated by CASCADE [11,12] and the results are discussed.

BACKGROUND

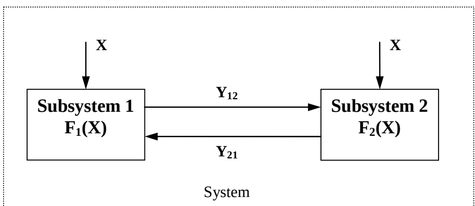

Fig. 1. MDO problem schematic.

Figure 1 shows the schematic of a tightly coupled MDO problem with two subsystems, where **X** represents the system level design variable vector, while $\mathbf{F_1}$ and $\mathbf{F_2}$ are the local objective functions for subsystem 1 and subsystem 2, respectively. Each subsystem can also have its own set of constraints. The $\mathbf{Y_{ij}}$ variables represent the behavior variables or coupling variables being passed from subsystem i to subsystem j. The subsystem output **Y**, is a function of local design variables and coupling variables from other subsystems, as represented in equation (1).

$$\mathbf{Y_{12}=f(X, Y_{21})}$$

$$\mathbf{Y_{21}=f(X, Y_{12})} \quad (1)$$

Figure 2 illustrates a multidisciplinary design synthesis based MDF formulation and derivative based optimization. The process starts with an initial design point. System Analysis is carried out on the current design point to find converged values of the coupling variables. It is followed by a sensitivity analysis step to compute total derivative information required to form local approximations to be used in the optimization step. The process repeats iteratively going from one design point to other until an optimum is found.

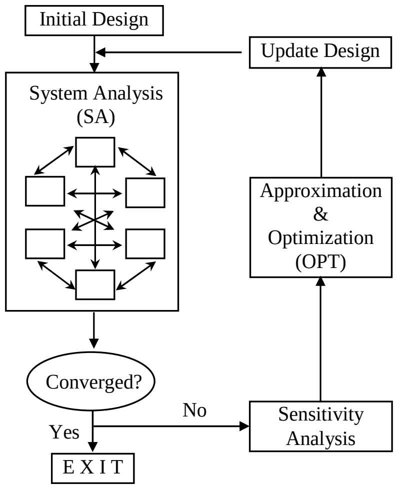

Fig. 2. MDO design synthesis.

Other MDO formulations such as IDF, AAO, and CO do not require SA but increase the problem size with additional equality constraints to handle coupling variables. In either approach the solution process is computationally expensive. The MDO formulations were initially developed for single system level objective function. Extending a similar derivative based approach to address multiple objectives in an MDO scenario would require several optimization runs in order to generate the complete Pareto frontier. In the next section a few of the recent MDO methods that solve for multiple objectives are discussed.

Multi-objective MDO Methods

Few of the recent MDO methods that have been proposed in the literature include Multi-Objective Pareto Concurrent Subspace Optimization (MOPCSSO) [13-16], Multi-Objective Collaborative Optimization (MOCO) [17,18], Multi-Objective Multidisciplinary Genetic Algorithm (M-MGA) [19], Interactive

Multi-Objective Optimization Design Strategy for Decision Based Multidisciplinary Design (iMOODS with MDO capability) [20], and Multi-Objective Genetic Algorithm Concurrent Subspace Optimization (MOGACSSO) [21]. MOPCSSO, MOCO and iMOODS can only generate one Pareto point after the system convergence. Generating the complete Pareto frontier requires multiple system convergence or multiple designer interactions. Besides, MOCO has the same disadvantage as CO where the problem size increases dramatically as the coupling density increases. M-MGA can generate multiple Pareto points fairly quickly but is only applicable to hierarchical MDO problems that could be decomposed into one system level sub-problem and one or more subsystem level sub-problems. M-MGA does not address the tightly coupled MDO problems that involve expensive system analysis. M-MGA also requires that the objectives be separable or additively separable [19]. MOGACSSO is able to generate the complete Pareto frontier for tightly coupled MDO problems, but require additional steps of allocating design variables. It could be beneficial for systems with a large number of design variables. The approach proposed in this paper is designed to be quite straightforward without a complicated formulation.

Genetic Algorithms for Multi-Objective Optimization

Genetic Algorithm (GA) based multi-objective optimization methods are particularly interesting because of their ability to generate a relatively uniform, reasonably well-distributed Pareto frontier fairly quickly and efficiently. Another advantage of using GAs for multi-objective optimization is their ablilty to generate non-convex and even discontinuous Pareto frontiers, without requiring any prior knowledge of the Pareto frontier or utopia point. The Vector Evaluated Genetic Algorithm (VEGA) [22], the Non-dominated Sorting Genetic Algorithm (NSGA) [4,23], the MOGA with rank-based fitness assignment by Fonseca and Fleming [24], and the MOGA with improvements by Narayanan and Azarm [25] are few of the many variants of MOGA that have been developed. These various MOGAs differ from each other with respect to fitness assignment, ranking scheme, selection method, and stopping criteria. Extensive literature is available on Evolutionary multi-objective optimization methods, including the MOGA [26,27] approach. The work developed in this paper uses Non-dominated Sorting Genetic Algorithm II (NSGA-II) of prof. K. Deb et al. [4], as implemented in modeFRONTIER. NSGA-II in modeFRONTIER provides a good diversity and spread of solutions and implements elitism for the multi-objective search.

MULTI-OBJECTIVE MDO SOLUTION WITH modeFRONTIER

The two main reasons for presenting the multi-objective MDO solution strategy using modeFRONTIER are 1. to show the ease of generating the complete Pareto frontier for a tightly coupled multi-objective MDO problem using a GA – based optimizer and 2. to do so without performing the expensive SA too often or increasing the problem size with a complex formulation. If a GA-based optimizer is to be used to solve a coupled MDO problem, it would require system analysis to be carried out for each member of the population and at every generation. Knowing that the system analysis phase is the greatest source of computational expense while solving a coupled MDO problem, such an approach would greatly undermine the effectiveness of using a GA for coupled MDO problem. To reduce the computational expense of system analysis, response surface approximations are created for objective functions and constraints using a Neural Network (NN). These NN approximations for objective functions and constraints are used during the NSGA-II optimization. In order to train the NN initially, full system analysis is carried out on a small population of designs, and true objective functions and constraints are evaluated as functions of design variables and coupling variables. This truly evaluated data set, also called the “Real” designs data set in modeFRONTIER is used to train the NN where design variables are treated as inputs while objective functions and constraints are treated as outputs. The NN approximation is also generated within modeFRONTIER using the RSM utility.

Once the NN approximations are created, the multi-objective problem is optimized using NSGA-II which utilizes NN approximations for the evaluation of objectives and constraints. The resulting converged designs are termed “virtual” in modeFRONTIER as they are evaluated using the NN response surface. From the virtual population, the Pareto and near Pareto designs are filtered and validated using real evaluation. Since a small population is used initially to train the NN, the response surface may not be highly accurate. Hence the real evaluation of the Pareto and near Pareto designs may not exactly match the virtual values. The newly calculated real values of these Pareto and near Pareto designs are used to retrain the NN. The NSGA-II optimization is performed again with the new NN response surface and the process is repeated until convergence. The accuracy of the NN approximation improves with subsequent filtering and training. With every cycle the population converges close to the real Pareto front with function values (objectives and constraints) falling in a narrow range and hence allowing improved NN approximations. Figure 3 shows the flowchart of the solution strategy. The steps are also discussed in the next section while describing the application of solution strategy to the MDO test problem.

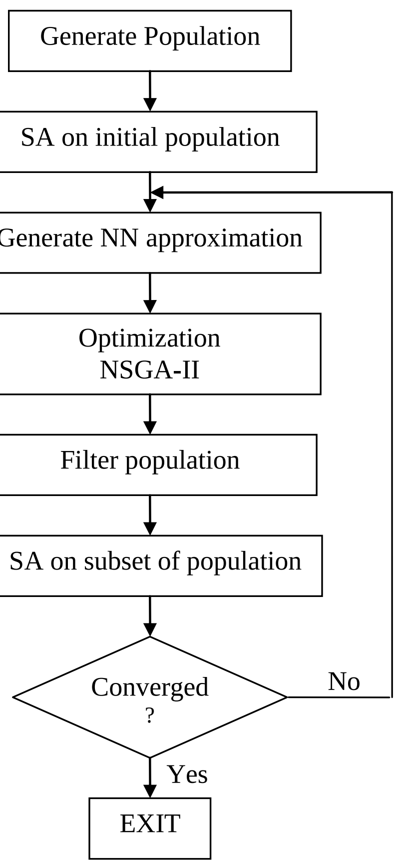

Figure 3. Flowchart of multi-objective MDO solution strategy using modeFRONTIER.

APPLICATION TO TEST PROBLEM

The solution strategy proposed in this paper is applied to a mathematical MDO test problem generated by CASCADE. CASCADE is classified as a generator of class 1 test problems (useful for testing and evaluating new MDO methods) by the NASA Langley Research Center MDO Test Suite (http://mdob.larc.nasa.gov/mdo.test/Problems.html). The test problem used in this paper consists of 2 subsystems, 2 objective functions, 10 design variables, 4 behavior variables, and 10 constraints, as shown in Figure 4. All the design variables are considered as system level design variables.

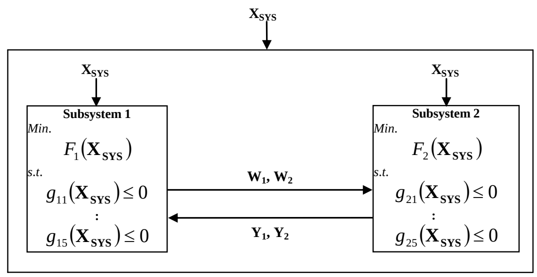

Figure 4. Test problem.

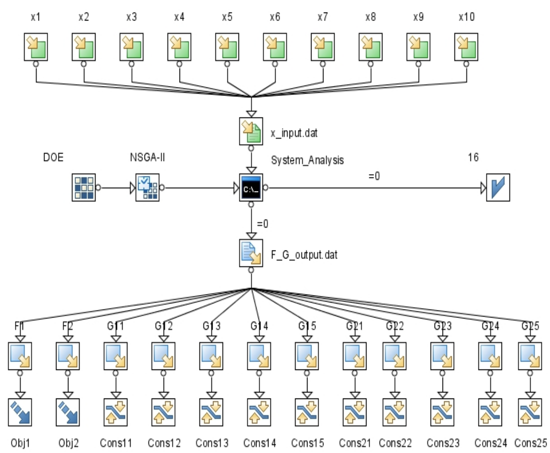

Figure 5. Problem setup in modeFRONTIER.

Figure 5 shows the workflow of the problem setup as established in modeFRONTIER. To perform the system analysis, a separate C code was written to perform the iterative convergence of coupling variables. The C code was integrated in the workflow as executable file. This C program is executed for each design variable while performing the real evaluation of objectives and constraints. A Matlab program could also be implemented instead of the C program. For systems involving complex solvers for structural and fluid analysis, respective softwares (solvers) could also be integrated in the workflow and solved iteratively.

Step 1: Initial population generation

Following the steps illustrated in Figure 3, initially a population of 200 designs is generated. The initial population can be generated either randomly or using one of the DOE algorithms available in modeFRONTIER. For this problem the SOBOL DOE algorithm was used.

Step 2: NN approximation

The initial population is evaluated with real analysis and used as training set to form the NN approximations.

Step 3: Optimization NSGA-II

Following this, the problem is optimized using NSGA-II while evaluating the designs virtually using the NN response surface. For optimization of initial cycles a population of 200 is used and optimized over 100 generations. For later cycles, after the NN is retrained, the population size is decreased and the number of generations is increased. Since function evaluation based on response surface is quite fast, a higher number of generations can be used for optimizations. NSGA-II optimization as implemented in modeFRONTIER incorporates elitism and also stores designs from previous generations in the database, allowing the generation of several Pareto designs without having to start with a large initial population. Figure 6 shows all the designs in performance space after optimization in first cycle. Gray designs are feasible, while orange are infeasible.

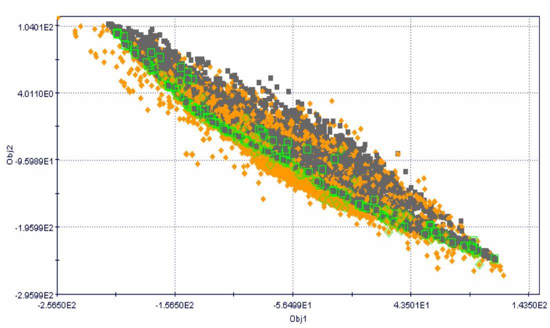

Figure 6. Performance space after 1st optimization.

Step 4: Filter population

After optimization, the Pareto and near Pareto designs are filtered from the database to check for convergence. In this case the last 200 designs from the database are selected. The selected designs are also highlighted in green in Figure 6. It can be seen that most of the designs are Pareto or near Pareto and are evenly distributed across the frontier. Alternatively, the designs can also be selected interactively from the performance space plot by dragging the mouse and highlighting desired points.

Step 5: Check convergence

The filtered designs are evaluated with real analysis. If the NN approximations created in step 2 are not accurate, the real analysis values of objectives and constraints for the filtered designs will not match very well with the virtual values. If the real values don't match well with the virtual values, the NN is retrained using the real evaluation of filtered designs as training data set. Figure 7 shows the real values of designs filtered in step 4 (shown selected in green in Figure 6) along with virtual values. Designs denoted with a "+" are real and infeasible. From Figure 7 it can be seen that the real and virtual values of the objective functions match quite well, but most of the designs which were feasible in virtual evaluation are not feasible in real evaluation. This means that the response surface for the objective functions is accurate to a good degree whereas the response surface for constraints is not. Since a retraining of NN is required, the cycle is repeated from step 2. Once the real and virtual values of designs match for the filtered designs after step 4, the solutions is assumed to have converged and the process is stopped.

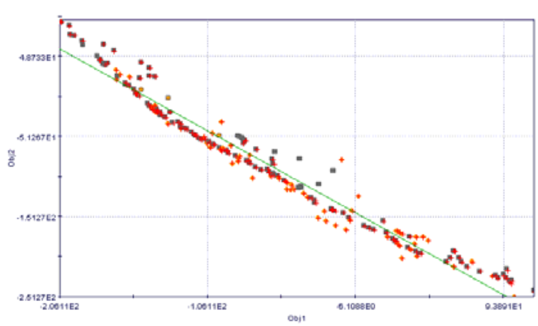

Figure 7. Comparing real and virtual designs.

For this problem the convergence criterion is met after four cycles. Figure 8 shows the final virtual population after optimization in fourth cycle. Figure 8 also shows the population in performance space. The gray designs are the feasible designs while orange designs are infeasible designs. The feasible and Pareto designs from the set are marked in green. The real evaluation of Pareto designs matched very well (for performance as well as constraints) with the virtual evaluation.

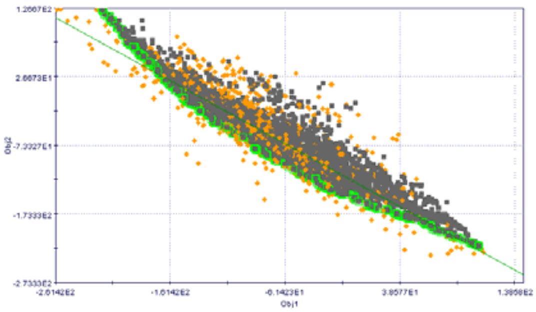

Figure 8. Optimization results after fourth cycle.

Figure 9 shows the final Pareto frontier after the fourth cycle. In Figure 9, the real and virtual designs obtained from modeFRONTIER after the fourth cycle are also shown overlapped on the Pareto frontier obtained from MOPCSSO. It can be seen that the Pareto designs generated by the technique developed in this paper are able to cover the entire Pareto frontier with a uniform distribution.

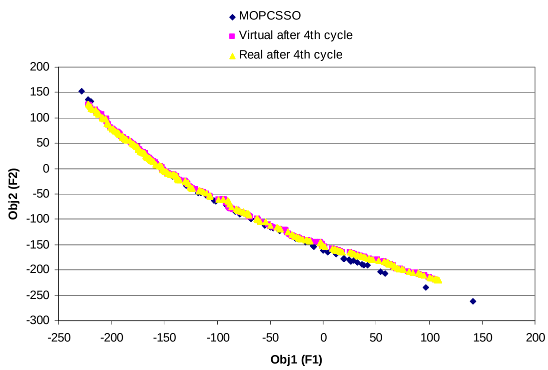

Figure 9. Pareto frontier after fourth cycle.

MOPCSSO required approximately 20 to 40 cycles to converge on one Pareto point (depending on the initial point). Hence, to populate 200 Pareto points, MOPCSSO would require on the order of 4000 to 8000 full system analysis (real function evaluations). In the approach presented here, a full system analysis is only performed on a subset of points (approximately 200 points were used in this work at every cycle to train the NN, although this could be reduced). Knowing that the current approach only required 4 cycles for convergence, approximately 800 full system analysis implementations (real function evaluations) were required to get 200 or more Pareto points. Hence, the approach presented in this paper can be said to be an order of magnitude more efficient than MOPCSSO.

CONCLUSION

In this paper a simple GA-based heuristic solution strategy is developed to solve a tightly coupled multi-objective MDO problem. The solution strategy is also shown to be easily implemented in modeFRONTIER. The solution technique is also shown to be able to solve a moderate sized coupled MDO problem with multiple objectives. The method can generate a fair number of Pareto designs with an even distribution along the complete Pareto frontier. The technique is shown to be quite efficient compared to other methods while still utilizing a simple problem formulation. It can also be seen that modeFRONTIER provides all the necessary tools which could be coupled very easily in a single software environment.

The computational expense of system analysis is reduced by creating Neural Network-based response surfaces and utilizing them during optimization. Neural Network is not the only choice of response surface, other response surface algorithms, such as Radial Basis Functions and Kriging, could also be utilized. Further savings could be achieved during retraining of response surface by retraining only those functions (objectives or constraints) which show considerable variation between real and virtual values at every cycle.

The approach presented here could also be easily extended to consider variation on design parameters and search for robust design solutions.

REFERENCES

[1] Multi-objective Optimization Design Environment www.modefrontier.com

[2] Cramer, E. J., Frank, P. D., Shubin, G. R., Dennis, J. E., and Lewis, R. M., 1992, On Alternative Problem Formulations for Multidisciplinary Optimization. Proceedings of 4th AIAA/USAF/NASA/OAI Symposium on

Multidisciplinary Analysis and Optimization, Cleveland, OH, AIAA 92-4752.

[3] Cramer, E. J., Dennis, J. E., Frank, P. D., Lewis, R. M., and Shubin, G. R., 1994, Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization*, **4**(4), 754-776.

[4] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. 2000, A Fast and Elitist Multi-Objective Genetic Algorithm-NSGA-II, *KanGAL Report Number 2000001*

[5] Sobieszczanski-Sobieski, J. , 1982, A Linear Decomposition Method for Optimization Problems – Blueprint for Development. *NASA TM 83248*.

[6] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., 1994, Multidisciplinary Optimization Methods for Aircraft Preliminary Design. Proceedings of 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama, FL, AIAA 94-4325.

[7] Braun, R.D., Kroo, I.M., and Moore, A. A., 1996, Use of the Collaborative Optimization Architecture for Launch Vehicle Design. Proceedings of 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, AIAA 96-4018.

[8] Sobiesezczanski-Sobieski, J., 1998, Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems. Proceedings of 2nd NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Hampton, VA.

[9] Bloebaum, C. L., Hajela, P. and Sobieszczanski-Sobieski, J., 1992, Non-Hierarchic System Decomposition in Structural Optimization. *Engineering Optimization*, **19**(3), 171-186.

[10] Renaud, J. E., and Gabriele, G. A., 1993, Improved Coordination in Non-Hierarchic System Optimization. *AIAA Journal*, **31**(12), 2367-2373.

[11] Hulme, K. F., and Bloebaum, C. L., 1997, Development of a Multidisciplinary Design Optimization Test Simulator. *Structural Optimization*, **14**(2-3), 129-137.

[12] Hulme, K. F., 2000, The Design of a Simulation-Based Framework for the Development of Solution Approaches in Multidisciplinary Design Optimization. Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Buffalo, NY.

[13] Huang, C. H., and Bloebaum, C. L., 2004, Multi-Objective Pareto Concurrent Subspace Optimization for Multidisciplinary Design. Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada.

[14] Huang, C. H., and Bloebaum, C. L., 2004, Visualization as a Solution Aid for Multi-Objective Concurrent Subspace Optimization in a Multidisciplinary Design Environment. proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY.

[15] Huang, C. H., and Bloebaum, C. L., 2004, Incorporation of Preferences in Multi-Objective Concurrent Subspace Optimization. proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY.

[16] Huang, C. H., 2003, Development of Multi-Objective Concurrent Subspace Optimization and Visualization Methods for Multidisciplinary Design. Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, Buffalo NY.

[17] Mc Allister, C., Simpson, L. T., Lewis, K., and Messac, A., 2004 Robust Multiobjective Optimization Through Collaborative Optimization and Linear Physical Programming. Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York.

[18] Tappeta, R.V., and Renaud, J., 1997, Multiobjective Collaborative Optimization. *Journal of Mechanical Design*, **119**, 403, 411.

[19] Gunawan, S., Azarm, S., Wu, J., and Boyars, A., 2003, Quality-Assisted Multi-Objective Multidisciplinary Genetic Algorithms. *AIAA Journal*, **41(**9), 1752, 1762.

[20] Tappeta, R. V., Renaud, J. E., and Rodriguez, J. F., 2002, An Interactive Multi-Objective Optimization Design Strategy for Decision Based Multidisciplinary Design. *Engineering Optimization*, **34**(5), 523, 544.

[21] Parashar, S., and Bloebaum, S., 2006, Multi-Objective Genetic Algorithm Concurrent Subspace Optimization (MOGACSSO) for Multidisciplinary Design. Proceedings of the 11th AIAA MDO Specialist Conference, Newport, RI.

[22] Schafer, J. D., 1985, Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. proceedings of the 1st International Conference on Genetic Algorithms.

[23] Srinivas, N., and Deb, K., 1995, Multiobjective Optimization Using

Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, **2**(3), 221, 248.

[24] Fonseca, C. M., and Fleming, P. J., 1993, Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization. proceedings of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, CA.

[25] Narayanan, S., and Azarm, S., 1999, On Improving Multiobjective Genetic Algorithms for Design Optimization. *Structural Optimization*, **18**, 146, 155.

[26] Fonseca C. M., and Fleming, P. J., 1995 An Overview of Evolutionary Algorithms in Multi-Objective Optimization. *Evolutionary Computation*, **3**, 1, 18.

[27] Deb, K., 2001, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK.