

# Topographical mapping assisted Evolutionary Search for Multilevel Optimization

Mohammed A. El-Beltagy and Andy J. Keane

*Computational Engineering and Design Centre*

*University of Southampton*

*Highfield, Southampton SO17 1BJ*

*United Kingdom*

*{M.A.El-Beltagy, Andy.Keane}@soton.ac.uk*

## Abstract

In many problems in science and engineering, it is often the case that there exist a number of computational models to simulate the problem at hand. These models are usually trade-offs between accuracy and computational expense. Given a limited computation budget, there is need to develop a framework for selecting between different models in a sensible fashion during the search. The method proposed here is based on the construction of a heteroassociative mapping to estimate the differences between models, and using this information to guide the search. The proposed framework is tested on the problem of minimizing the transmitted vibration energy in a satellite boom.

## 1. Introduction

In the optimization literature there is generally a tendency to neglect the issue of finite computational resources. Evolutionary based search often requires a substantial number of function evaluations to find a good optimum. Hence, simplified computational models are commonly used for conducting such a search. This approach, while justifiable on practical grounds, may miss out on good optima that only exist in more complex models. In some cases it might even lead to a false optimum (i.e. one that doesn't exist on the complex model).

The multiplicity of computational models for a given object of simulation may arise due to three main causes. It could be due to different mathematical formulations being used to construct the model such as Euler and Navier Stokes in computational fluid dynamics (CFD). It could also be due to different discretization limits with one formulation such as mesh densities in finite elements analysis (FEA). Finally, it may come from the availability of approximate empirical models such as neural networks or response surfaces.

We use here the term "multilevel optimization" (MLO) to denote the process of optimizing such a multiplicity of models, where each level is essentially one of these models.

In contrast to the static optimization problem

$$f(x) \rightarrow \text{opt}, (x \in M),$$

the multilevel optimization problem may be stated as

$$f_1(x) \rightarrow \text{opt}, (x \in M),$$

where  $f_1(x)$  is the most accurate function and there exist many  $f_k(\bar{x})$  models where  $k = 1..L$ .

The levels are such that  $f_i(x)$  is more accurate and computationally expensive than  $f_j(x)$  for  $i < j$ .

During the optimization, it is only feasible that the computationally cheap function levels be used often. At the same time, results obtained from evaluations on the more accurate levels should be used to guide the search on the less accurate ones.

For simplicity, we consider in this paper the case of just two levels  $f_e(x)$  and  $f_a(x)$ . Here  $f_e(x)$  represents the exact and computationally expensive model and  $f_a(x)$  the approximate and computationally cheap one.

Our proposed approach works by building a heteroassociative mapping to estimate the agreement between the exact and approximated evaluation for given  $x$ . It partitions the search space into a discrete number of areas with a discrepancy estimate for each area. This information is then used for model selection.

The paper is organized as follows: in the next section we give a brief overview of previous work relating to multilevel optimization. Section 3 describes the Generative Topographical Mapping (GTM). Section 4 describes the GTM-EA synthesis for MLO. Section 5 discusses the satellite boom problem and the results obtained. The paper closes with a brief conclusion and discussion of future work.

## 2. Previous work relating to MLO

Dunham *et al.* were first to address the problem of multilevel optimization within an evolutionary optimization context [5]. They worked with a two level problem. In their study they used an approximate model most of the time, using the accurate/computationally expensive model only at the final stages of refinement. They state

It seemed better to run through many "generations" with only approximate scores indicating progress than to manage a very few "evolutions" with rather exact statements of position.

Later efforts focused on building variants of injection island genetic algorithms (iiGA) architectures for this sort of optimization [15],[7],[6]. The approach adopted was to have many islands using low accuracy/cheap evaluations that progressively pass on individuals to fewer islands using higher accuracy/expensive evaluations.

A case for the importance of multilevel optimization for aeronautical design was presented in [14]. [2] presents fundamental studies focusing on the use of a family of approximate fitness representations in GAs. A computational framework for integrating a class of single-point approximation models with GAs was proposed in [13]. More recently, this framework for coevolutionary GAs applied to the design of large flexible space structures was extended in [12].

## 3. The Generative Topographic Mapping (GTM)

The GTM is a probabilistic re-formulation of Kohonen's self-organizing map (SOM) [10] which avoids many of its theoretical limitations. The GTM is an efficient method for vector quantization. Its power lies in capturing and representing, in a lower dimensional space, the important features from a high dimensional space. It is a probability density model which describes the distribution of data in space of several dimensions in terms of two 'latent' variables corresponding to the coordinates of the latent space. A detailed derivation of the GTM is beyond the scope of this paper, but can be found in [4]. Here we simply sketch its essential features.

The non-linear mapping from the two dimensional latent space  $\mathbf{u}$  to the  $D$  dimensional data space  $\mathbf{t}$  takes the form

$$\mathbf{y}(\mathbf{u}) = \mathbf{W}\phi(\mathbf{u}), \quad (1)$$

where  $\phi = (\phi_1, \dots, \phi_M)^T$  represents a set of  $M$  non-linear basis functions, and  $\mathbf{W}$  is  $D \times M$  matrix of parameters. The mapping in (1) defines a two-dimensional non-Euclidean manifold  $S$  embedded in the  $D$ -dimensional Euclidean data spaces. The basis functions used here are a set of Gaussians centered on a square grid in latent space.

Since the data will not live exactly on such a manifold, the distribution of the data around the manifold takes the form

$$p(\mathbf{t}|\mathbf{u}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{u}; \mathbf{W}) - \mathbf{t}\|^2\right\}. \quad (2)$$

The distribution in  $\mathbf{t}$ -space, for a given values of  $\mathbf{W}$  and  $\beta$ , is obtained by integration over the  $\mathbf{u}$ -distribution

$$p(\mathbf{t}|\mathbf{W}, \beta) = \int p(\mathbf{t}|\mathbf{u}, \mathbf{W}, \beta)p(\mathbf{u})d\mathbf{x}. \quad (3)$$

The GTM algorithms corresponds to a particular form of this model in which  $p(x)$  is considered to be a sum of delta functions centered on the nodes of the rectangular lattice in the latent space

$$p(x) = \frac{1}{K} \sum_{l=1}^K \delta(u - u_l) \quad (4)$$

This lattice is typically much finer than the grid of points used to define the centers of the basis function (i.e.  $K \gg M$ ). Each point  $u_l$  is mapped to the corresponding point  $y(u_l; W)$  in the data space, which forms the center of a Gaussian density function. From (3) and (4) the distribution function in data space takes the form

$$p(t|W, \beta) = \frac{1}{K} \sum_{l=1}^K p(t|u_l, W, \beta). \quad (5)$$

The above form represents a mixture of Gaussians in which the centers of the Gaussian functions are constrained to lie on the two-dimensional manifold  $S$ .  $W$  and  $\beta$ , are determined by the maximum likelihood using the EM (expectation-maximization) algorithm [3].

The node locations in the data space (also called 'reference vectors') as defined by (1) are given by

$$m_i = W\phi(u_i) \quad (6)$$

Besides being an effective vector quantization tool the GTM, like its predecessor the SOM, can be used for visualization and data mining purposes [8].

The GTM leads to an additional memory and computational burden that can only be justified if the models considered are expensive enough.

#### 4. Topographical mapping assisted Evolutionary Search

We use the GTM to partition the search space into number of areas with different estimates of the discrepancies between the exact model  $f_e(x)$  and the approximate one  $f_a(x)$ . We are hence using it as a *look-up table* that is continuously updated as the search progresses.

We denote the set of data vectors used to construct the GTM by  $T_N \equiv \{t_i\}_{i=1}^N$ . Each data vector takes the form

$$t_i = [x_i, |f_e(x_i) - f_a(x_i)|] \quad (7)$$

During training only the  $x_i$  part of  $t_i$  is used in the node *responsibility* calculations in the E-step in order to construct an asymmetric (heteroassociative) mapping [10, p. 259].

Given a new  $x_j$ , the trained GTM is used to estimate  $|f_e(x_j) - f_a(x_j)|$ . The closest reference vector  $m_c$  to  $x_j$  using the Euclidean distance measure  $\|x_j - m_i\|$  is found as shown in (8). The distance is determined on the basis of the  $x$  part only. The  $|f_e(x_j) - f_a(x_j)|$  component of  $m_c$  is the estimate.

$$\|x_j - m_c\| = \min_i \{\|x_j - m_i\|\} \quad (8)$$

At any instance during the search there is current tolerance level  $\varepsilon_{curr}$  which determines how large  $|f_e(x_j) - f_a(x_j)|$  would be allowed before an accurate evaluation is made. If  $|f_e(x_j) - f_a(x_j)| > \varepsilon_{curr}$ , then  $f_e(x)$  is used for evaluation, otherwise  $f_a(x)$  is used.

Given a limited computational resource that is equivalent to  $N_e$  accurate function evaluations, a proposed strategy for controlling  $\varepsilon_{curr}$  is given by

$$\varepsilon_{curr} = \varepsilon_{\max} \left( 1 - \frac{(n_e + \eta_{ale} n_a)}{N_e} \right), \quad (9)$$

where  $\varepsilon_{\max}$  is the maximum allowable tolerance,  $n_e$  is the number of exact evaluations carried so far,  $n_a$  is the number of approximate calculations, and  $\eta_{ale}$  is the ratio in computational effort between an approximate and

exact evaluation. The above formula provides a linear decrease in  $\varepsilon_{curr}$ , and as the search progresses, less and less individuals are evaluated using  $f_a(x)$ . This is done to ensure more aggressive use of the exact model during the final stages of the search.

```

 $n_e = 0$ 
 $n_a = 0$ 
Begin
  Initialization of population
  Calculate  $|f_e(x) - f_a(x)|$  for all members to construct initial  $T_N$ 
  Build initial GTM using  $T_N$ 
  while  $\left(1 - \frac{(n_e + \eta_{a/e} n_a)}{N_e}\right) > 0$ 
    Apply evolutionary operators
    for every member of the population
      if  $|f_e(x_j) - f_a(x_j)| > \varepsilon_{curr}$ 
        evaluate using  $f_e(x)$ 
        evaluate using  $f_a(x)$ 
         $n_e = n_e + 1$ 
         $n_a = n_a + 1$ 
        Calculate  $|f_e(x) - f_a(x)|$  and update  $T_N$ 
      else
        evaluate using  $f_a(x)$ 
         $n_a = n_a + 1$ 
      end if
    end for
    Do elitism and reevaluate the best individual if necessary
     $\varepsilon_{curr} = \varepsilon_{max} \left(1 - \frac{(n_e + \eta_{a/e} n_a)}{N_e}\right)$ 
    Reconstruct GTM using the expanded  $T_N$ 
  end while
End

```

The general structure of the algorithm is shown above. First the initial population is created and evaluated using both  $f_e(x)$  and  $f_a(x)$ . The initial GTM model is then built based on the initial population evaluations. When making subsequent evaluations, if the tolerance criterion is not met, both  $f_e(x)$  and  $f_a(x)$  are evaluated and the GTM training  $T_N$  is expanded. The individual's score takes on the  $f_e(x)$  value.

To prevent the EA from heading towards a false optima while using the approximate model, elitism is employed. In this scheme, at the end of every generation the best individual is reevaluated using  $f_e(x)$  (that is if it wasn't already evaluated using  $f_e(x)$  due to the tolerance criterion) and then compared to the last best.

At the end of every generation the tolerance level is updated and the GTM is rebuilt using the expanded  $T_N$ .

The stopping criterion  $\left(1 - \frac{(n_e + \eta_{a/e} n_a)}{N_e}\right) > 0$  is designed such that optimization will cease when the computational cost equivalent of  $N_e$  accurate function evaluations has been carried out.

The GTM's weights adapt and change during the search; in a sense they could be regarded as Population-level adaptive parameters [1].

## 5. Experiments on Satellite Boom Optimization

A two dimensional satellite boom structure was used to evaluate the proposed method. The baseline structure is shown below in Figure 1. It consists of 40 individual Euler-Bernoulli beams connected at 20 joints [9]. Each of the 40 beams has the same properties per unit length. The beam is excited at one end as shown. The goal of the optimization was set at minimizing the frequency averaged response of the end beam in the range 150-250 Hz. The optimizer was allowed to generate new geometries by varying the coordinates of the inner 18 joints of this simplified structure. That is 36 optimization variables representing the  $x$  and  $y$  joint positions.

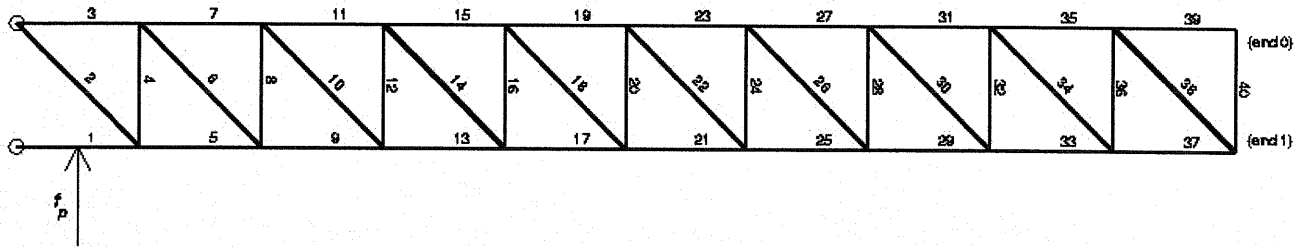


Figure 1 The 2D beam baseline structure

A typical frequency response is shown below in Figure 2. In the figure the un-optimised response refers to baseline structure as shown in Figure 1. The optimized structure for this particular graph is shown in Figure 3.

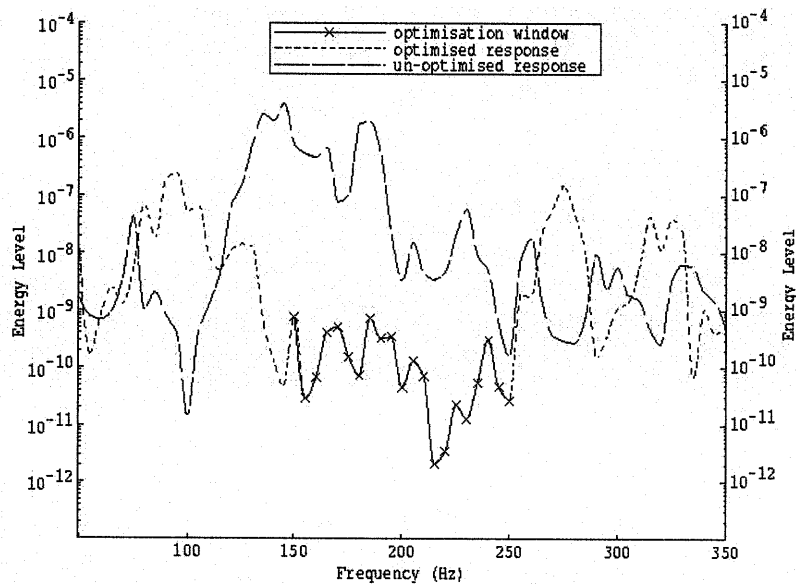


Figure 2 Frequency response of the 2D beam structure.

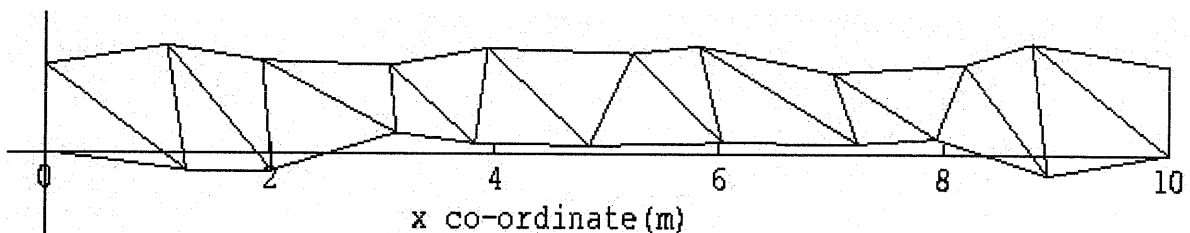


Figure 3 Optimized 2d beam .

In this problem, the objective function to be minimized depends on the area under the graph in Figure 2 in the range of 150-250 Hz. The calculation of a single frequency point in this plot requires the solution of some 260 complex simultaneous equations and this takes about 2.3 seconds on Silicon Graphics Challenge machine with R10K chips. The more points that are used to perform the integration, the more accurate are the evaluations. For our 2-level study, we used 11 points for exact evaluation and 5 for the approximate. Figure 4 below shows a 2D section of the log of the objective function using 11 and 5 integration points. It is clear from the figure that there are some areas where both models agree and areas where they disagree.

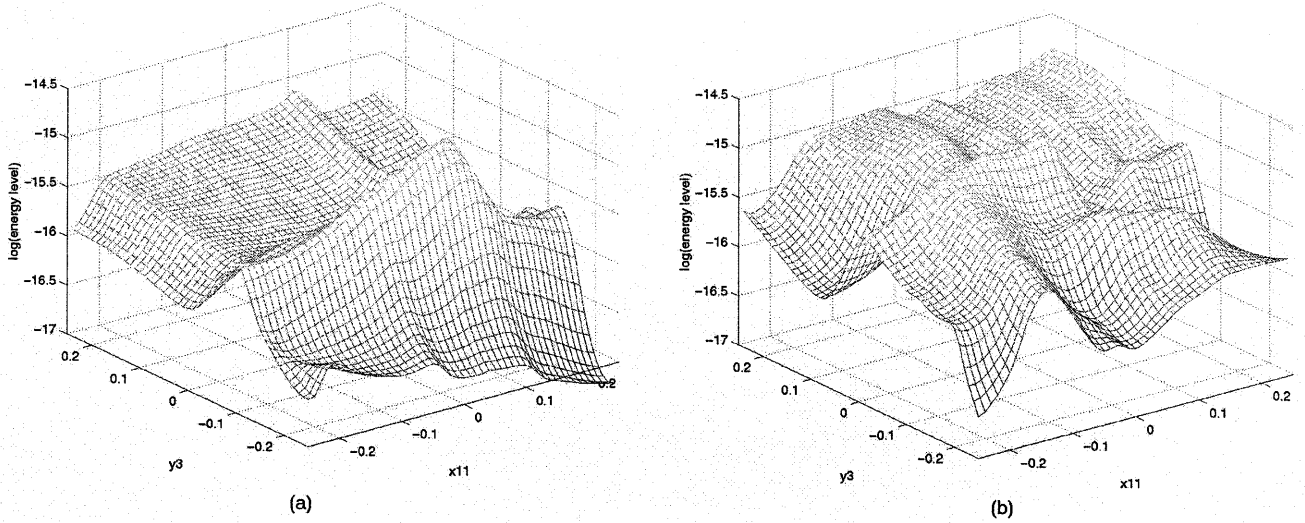


Figure 4 A two dimensional cross section of the log on the energy level over the 150- 250 Hz range using (a) 5 integration points and (b) 11 integration points

In this problem, the inaccurate evaluation can be readily obtained from the accurate evaluation by simply using a fewer number of integration points to get the approximate value. Therefore the stopping criterion and tolerance update were modified to

$$\left(1 - \frac{(n_e + \eta_{a/e}(n_a - n_e))}{N_e}\right) > 0, \quad (10)$$

$$\text{and } \varepsilon_{curr} = \varepsilon_{max} \left(1 - \frac{(n_e + \eta_{a/e}(n_a - n_e))}{N_e}\right). \quad (11)$$

We used a GA for the experiments reported here. It is typical of those described in [11]. The GA makes use of arithmetic crossover, gaussian mutation, and the tournament selection operator. The crossover and mutation probabilities were kept constant at 0.9 and 0.15 for all the results reported here. The population size was set to 100, and the tournament size was set to 2. The GTM used had 49 nodes and 16 basis function centers arranged on a square lattice. The computational load  $N_e$  limit was set to 1500.

Three sets of runs were executed to assess the performance of the GTM-EA. In each set ten runs were performed and averages taken.

In the first set the GTM-EA approach was used as described in section 4. To give a basis of comparison the second set of results use the same ratio of accurate to approximate function call as in the first set. However, in this case all the approximate calls were used first in the GA and then the resulting population used to seed a second GA that exclusively called the accurate functions. In the table below we denote this approach by the term sequential.

Finally, a third set of runs was carried out where, instead of the GTM being built using the first population, the initial map was instead pre-computed using the data from the first set of runs. This of course means that in the population initialization of this set there is no need to use the accurate function evaluation for the members. Rather the accurate function was only used when needed by the algorithm.

	Average
GTM-EA	0.73
Sequential	1
GTM-EA with previously computed points	0.47

Table 1 Normalized energy levels

The above table shows the average energy level for each set of runs, normalized with respect to the average sequential runs. The improvements obtained by the GTM-EA synthesis over the simple sequential approach may be seen as not insignificant. Moreover, as the GTM acquires more data it becomes more capable at model selection and arrives at better results. Unlike the sequential approach, the GTM gets better with time as it gains more "experience". This provides an effective way of merging data from different runs on the same problem. Also the GTM can be of great assistance as a visualization tool in understanding the discrepancies between different models and hence could allow the user to reformulate the optimization problem.

## 6. Conclusion and future work

In this paper we have made the case for multilevel optimization under a limited computational budget. We have presented a novel approach to tackle this problem and have shown it to work on a real world engineering problem.

The motivation behind using topographical mapping first started with self organizing maps (SOM). The SOM was seen as a form of associative memory that can be used to store collective knowledge in an adaptive fashion. It shapes and is shaped by the process of evolution. Something akin to cultural traits. We then moved to GTM as a more principled alternative founded on more solid statistical principles.

Future research will focus on building statistical models that are more suited to this problem than GTM. The improvements are hoped to be more dramatic.

It is also hoped that a combination of the proposed approach with injection island GA (iiGA) for a two way "migration control" will result in an more efficient parallel GA for multilevel optimization.

## Acknowledgments

This work was supported under EPSRC grant no GR/L04733. Mohammed El-Beltagy is grateful to the organizers of the School on Natural Computation, 1997 for a very stimulating program, where the seeds of the ideas presented came into being. We would like to thank Markus Svensén for his assistance during our implementation of the GTM.

## References

- [1] P. J. Angeline, "Adaptive and Self-Adaptive Evolutionary Computations," in *Computational Intelligence: A Dynamic Systems Perspective*, M. Palaniswami, Y. Attikiouzel, R. Marks, D. Fogel, and T. Fukuda, Eds. Piscataway, NJ., 1995, pp. 152-163.
- [2] M. A. El-Beltagy and A. J. Keane, "Optimization for Multilevel problems: A Comparison of Various Algorithms," in *Proc. of The Third International Conference on Adaptive Computing in Design and Manufacture (ACDM '98)*, I. Parmee, Ed. London: Springer, 1998, pp. 111-120.
- [3] C. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford University Press, 1995.

- [4] C. Bishop, M. Svensen, and C. K. I. Williams, "GTM: The generative topographic mapping," *Neural Computation*, vol. 10, pp. 215-234, 1998.
- [5] B. Dunham, D. Fridshal, R. Fridshal, and J. H. North, "Design by Natural Selection," *Synthese*, vol. 15, pp. 254-259, 1963.
- [6] D. Eby, R. C. Averill, W. F. Punch, and E. D. Goodman, "Evaluation of Injection Island GA Performance on Flywheel Design Optimization," in *Proceedings of Third Conference on Adaptive Computing in Design and Manufacturing*, I. Parmee, Ed. London: Springer Verlag, 1998, pp. 121-136.
- [7] E. D. Goodman, R. C. Averill, W. F. Punch, and D. Eby, "Parallel Genetic Algorithms in the Optimization of Composite Structures," in *Second World Conference on Soft Computing (WSC2)*, 1997 .
- [8] S. Kaski, "Data Exploration Using Self-Organizing Maps," Doctoral Thesis, Helsinki University of Technology, 1997.
- [9] A. J. Keane, "Passive Vibration Control Via Unusual Geometries: The Application of Genetic Algorithm Optimization to Structural Design," *Journal of Sound and Vibration*, vol. 185(3), pp. 441-453, 1995.
- [10] T. Kohonen, *Self-Organizing Maps*, Second Edition ed. Berlin: Springer-Verlag, 1997.
- [11] Z. Michalewicz, *Genetic Algorithms+Data Structures =Evolution Programs*, 3rd ed. New York: Springer, 1996.
- [12] P. B. Nair, A. J. Keane, and R. S. Langley, "Design Optimization of Space Structures With Nonperiodic Geometries for Vibration Suppression," in *Proceedings of the 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Missouri: AIAA, 1999, pp. AIAA 99-1260.
- [13] P. B. Nair, A. J. Keane, and R. P. Shimpi, "Combining Approximation Concepts With Genetic Algorithm-Based Structural Optimization Procedures," *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures*, vol. 2, pp. 1741-1751, 1998.
- [14] G. M. Robinson and A. J. Keane, "A Case for Multi-level Optimization in Aeronautical Design," in *Proceedings of the Royal Aeronautical Society Conference on Multidisciplinary Design and Optimization.*, 1998.
- [15] H. D. Vekeria and I. Parmee, "Co-operative Evolutionary Strategies for Single Component Design," in *Proceedings of The Seventh International Conference on Genetic Algorithms*, T. Bäck, Ed. San Fransico, CA: Morgan Kaufmann, 1997, pp. 529-536.