Metamodeling Techniques For Evolutionary Optimization of Computationally Expensive Problems : Promises and Limitations

Mohammed A. El-Beltagy, Prasanth B. Nair and Andy J. Keane
Computational Engineering and Design Centre
University of Southampton
Highfield, Southampton SO17 1BJ
United Kingdom
E-Mail: {M.A.El-Beltagy, P.B.Nair, Andy.Keane}@soton.ac.uk

Abstract

It is often the case in many problems in science and engineering that the analysis codes used are computationally very expensive. This can pose a serious impediment to the successful application of evolutionary optimization techniques. Metamodeling techniques present an enabling methodology for reducing the computational cost of such optimization problems. We present here a general framework for coupling metamodeling techniques with evolutionary algorithms to reduce the computational burden of solving this class of optimization problems. This framework aims to balance the concerns of optimization with that of design of experiments. Experiments on test problems and a practical engineering design problem serve to illustrate our arguments. The practical limitations of this approach are also outlined.

1 INTRODUCTION

Evolutionary search on complex high dimensional, multi-modal parameter optimization problems often requires a very high number of function evaluations. In many practical situations, this computational expense cannot be afforded. A typical example is that of aerodynamic wing design, where each function evaluation may require a computational fluid dynamics (CFD) simulation costing hours of supercomputer time. Hence, when the difficulties of high computational cost are coupled with nonconvex design spaces, standard evolutionary optimization approaches may be impractical and more advanced strategies are required.

Metamodels are statistical models that are built to approximate detailed computer analysis codes (Simpson *et al.*, 1997). Metamodels are orders of magnitude cheaper to run, and can be used in lieu of detailed analysis during evolutionary search. The underlying premise of this approach is that, one can construct an *approximation* of the analysis codes that is much more efficient to run, and which yields insight into the functional relationship between the input variables $\boldsymbol{x}$ and the output y. If the true nature of a computer analysis code is represented as

$$y = f(\boldsymbol{x}), \tag{1}$$

then a metamodel or "model of the model" is of the form

$$\hat{y} = g(\boldsymbol{x}), \text{ such that } y = \hat{y} + \varepsilon \tag{2}$$

where ε represents the error of the approximation. There exists a variety of different approximation techniques one could use to construct a metamodel (e.g., least square regression, backpropagating artificial neural nets, etc.).

We are concerned here with deterministic analysis models (i.e. a given $\boldsymbol{x}$ will always yield a singular y value, not a distribution). A "statistically sound" method for constructing a metamodel in this case would be through the use of kriging models, also referred to as the Design and Analysis of Computer Experiments (DACE) models in the statistics literature (Simpson, 1998; Sacks, *et al.*, 1989), and Gaussian regression techniques in the neural networks literature (Gibbs, 1997). Kriging techniques originated in the geostatistics community in the 1960's; an account of the history of these technique can be found in Cressie (1991). The problem of constructing metamodels is closely coupled with the field of design of experiments (Sacks, *et al.*, 1989).

Approximation model management frameworks for pattern search algorithms have been proposed in the literature (Dennis and Torczon, 1997). This framework uses DACE approximation models to accelerate the optimization procedure, and theoretically guarantees that the search converges to a local optima. Successful applications of this approach to engineering design problems with explicit algebraic constraints have recently been presented in the literature (Booker, *et al.*, 1998).

The integration of approximation models with evolutionary algorithms is a research topic which has attracted some attention of late. Ratle (1998) presented an approach wherein kriging techniques were used for fitness landscape approximations in GAs. Encouraging results were presented for some multimodal test functions. However, the details of how the data points are selected for constructing and updating the kriging model are not reported. It is important to note here that the problem of designing experiments to generate numerical data for constructing a statistical metamodel is of crucial importance to the success of this approach.

Earlier efforts have focused on selecting between computationally expensive models and cheap approximation using variants of injection island genetic algorithms (iiGA). (Vekeria & Parmee, 1996; Goodman *et al.*, 1997; Eby *et al*., 1998)

A case for the use of variable-fidelity analysis models in evolutionary optimization was presented by Robinson and Keane, (1998). El-Beltagy and Keane (1998) presented 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 by Nair, *et al* (1998). More recently, Nair, *et al* (1999), extended this framework for coevolutionary GAs applied to the design of large flexible space structures.

In this paper, we present a general framework for integrating simulation metamodels with evolutionary optimization algorithms. The construction of a good metamodel requires an experimental design which is space-filling so as to capture the essential trends of the function landscape. In contrast, the goal of optimization is to generate points which lead to improvements in the objective function. We present an algorithm for adaptively selecting points for updating the metamodel, while balancing the often conflicting concerns of good experimental design with that of optimization. We also present experimental results on some test problems to outline the advantages and limitations of this approach.

This paper is organized as follows: in the next section we give a brief overview of Gaussian Regression Models. Section 3 describes the evolutionary-metamodeling synthesis. Section 4 details experiments results. The paper closes with a brief conclusion and discussion of future work.

2 GAUSSIAN REGRESSION MODELS

We are concerned with approximating deterministic computer models, and hence perfectly interpolating models are most germane to our concerns. This section presents a brief overview of the mathematical foundations of gaussian regression models, which are interchangeably referred to as Kriging and DACE models in this paper. This class of statistical models can be written as the combination of a polynomial model plus departures of the form

$$\hat{y}(\boldsymbol{x}) = p(\boldsymbol{x}) + Z(\boldsymbol{x}) \qquad (3)$$

where $\hat{y}(\boldsymbol{x})$ is the metamodel, $p(\boldsymbol{x})$ is a known polynomial function of $\boldsymbol{x}$, and $Z(\boldsymbol{x})$ is the realization of a normally distributed Gaussian random process with mean zero, variance σ^2, and non-zero covariance. The polynomial term $p(\boldsymbol{x})$ provides a global model while $Z(\boldsymbol{x})$ accounts for "localized" deviations. Here we use a constant term β for the global model $p(\boldsymbol{x})$ (Sacks, *et al*, 1998).

The covariance matrix of $Z(\boldsymbol{x})$ is given by

$$Cov\left[Z\left(\boldsymbol{x_i}\right), Z\left(\boldsymbol{x_j}\right)\right] = \sigma^2 C\left(\boldsymbol{x_i}, \boldsymbol{x}_j\right) \qquad (4)$$

where $R(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the spatial correlation function between sample points x_i and x_j from the set of sample points used to construct the model $\{\boldsymbol{x_i}\}_{i=1}^{N}$. The only constraint on the choice of a correlation function is that the resulting covariance matrix should be positive definite. A number of correlation functions suitable for deterministic computer experiments can be found in Sacks *et al* (1989).

A Gausian correlation function is employed in this work, which can be written as

$$R\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \exp\left(-\sum_{l=1}^{L} \theta_l \left(\boldsymbol{x}_i^{(l)} - \boldsymbol{x}_j^{(l)}\right)\right) \qquad (5)$$

where $x_i^{(l)}$ is the l^{th} component of $\boldsymbol{x_i}$, an L dimensional vector, and θ is the vector of correlation parameters.

Predicted estimates $\hat{y}(\boldsymbol{x})$, at untried values of $\boldsymbol{x}$ in the case when $p(\boldsymbol{x})$ is taken to be a constant term, can be written as

$$\hat{y}(\boldsymbol{x}) = \beta + \boldsymbol{k}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{C}^{-1}(\boldsymbol{y} - \beta\boldsymbol{1}) \qquad (6)$$

where $\boldsymbol{C}$ is the correlation matrix constructed from the N sample points using (5), $\boldsymbol{1}$ is a column vector of ones of length N, $\boldsymbol{y}$ is a column vector of length N containing the output of the analysis model at the sample points, and

$\boldsymbol{k}(\boldsymbol{x})$ is a correlation vector of length N between $\boldsymbol{x}$ and the sampled data points $\{\boldsymbol{x_i}\}_{i=1}^{N}$. It is given by

$$\boldsymbol{k}(\boldsymbol{x}) = \left(R(\boldsymbol{x_1}, \boldsymbol{x}), \ldots, R(\boldsymbol{x}_N, \boldsymbol{x})\right) \tag{7}$$

The model parameters β and θ are estimated by assuming that the function to be approximated is the realization of a gaussian stochastic process. This enables us to use the framework of Bayesian statistics to derive error bounds on the model predictions. Maximum Likelihood Estimation (MLE) is used to obtain the model parameters.

In (6) β is estimated as

$$\hat{\beta} = \left(\boldsymbol{1}^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{1}\right) \boldsymbol{1}^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{f} \tag{8}$$

where $\boldsymbol{f}$ is the vector of function values at the sample points.

The estimate of the variance of the sample point from the global model β is given by

$$\hat{\sigma}^2 = \frac{\left(\boldsymbol{f} - \boldsymbol{1}\hat{\beta}\right)^{\mathrm{T}} \boldsymbol{C}^{-1} \left(\boldsymbol{f} - \boldsymbol{1}\hat{\beta}\right)}{N}. \tag{9}$$

The correlation parameters are estimated by solving an unconstrained minimization problem which can be written as

$$\max_{\theta > 0} \frac{N \ln(\hat{\sigma}^2) + \ln(\det(\boldsymbol{C}))}{2} . \tag{10}$$

It can be clearly seen that DACE models are perfectly interpolating in nature. The reader is referred to the dissertation of Gibbs (1997), for a detailed overview of the computational implementation of DACE models and extensions to accommodate noisy data via non-interpolating models.

For the DACE model to work appropriately, the $\{\boldsymbol{x_i}\}_{i=1}^{N}$ sample points should be selected such that the trends of the underlying function can be captured effectively. To achieve this we need to select points which are space-filling in nature. However, if the sample points are too close together or identical, the correlation matrix may be ill-conditioned. Our metamodel update mechanism attempts to prevent this. In the present work, we also add a small diagonal term (10^{-6}) to the correlation matrix to circumvent this problem.

It has been shown by Neal (1996) that the properties of a neural network with one hidden layer converge to those of a Gaussian process as the number of hidden neurons tends to infinity if standard `weight decay' priors are assmued . This has motivated the idea of replacing supervised neural networks with Gaussian processes.

3 EVOLUTIONARY-METAMODELING SYNTHESIS

Evolutionary algorithms being population-based search techniques generate space-filling points as they explore the search space. The key idea of our algorithm is to select points for constructing the metamodel such that the concerns of good experimental design can be balanced with that of optimization. Once a baseline metamodel is constructed, it can be used in lieu of the computationally expensive analysis model for the fitness predictions. The points generated by the evolutionary algorithm are used to adaptively update the metamodel so as to improve the approximation of the search landscape and prevent the optimizer from getting trapped in a false optima within the metamodel.

```
Begin
  Random population initialization
  Evaluation of Np individuals
  while(number of accurate evalution <  maxeval)
    Apply evolutionary operators
    Evaluate using metamodel
    if(generation delay reached)
      Update metamodel
    end if
  end while
End
```

The general structure of our algorithm is shown above. The first step of the scheme involves building a baseline metamodel. There exists a number of experimental design techniques in the statistics literature (e.g., randomized orthogonal arrays, Latin hypercubes, etc.) which could be used to generate data for this purpose (Owen, 1992). In this research, we chose a subset from the randomly generated initial population such that they satisfy a minimum distance criterion.

We denote the set of vectors used to construct the metamodel by $\boldsymbol{X_N} \equiv \{\boldsymbol{x_i}\}_{i=1}^{N}$. The set of vectors of individuals in a given population is denoted by $\boldsymbol{P}_{Np} \equiv \{\boldsymbol{p_i}\}_{i=1}^{Np}$.

As shown, the proposed algorithm is structured around the maximum computational budget allowable for model evaluations. The parameter *maxeval* denotes the maximum number of model evaluations that can be used. Metamodel construction and evaluations are considered to be of negligible computational cost for the problems under consideration. We use the term *generation delay* to denote the frequency at which we update the metamodel.

```
Function Update metamodel
Begin
 Sort population in descending order
```

$$D = \left\{ \min_{x_j \in X_N} \left\| p_i - x_j \right\| \right\}_{i=1}^{Np}$$

```
for i=1 to Np*fitfac
  if( D_i >η)
    X_N = X_N ∪ p_i
    D_i = 0
  end if
end for
Sort D in descending order
for i=1 to Np*doefac
  if( D_i >ε)
    X_N = X_N ∪ p_{l_i}
    D_i = 0
  end if
end for
evaluate new inclusions of X_N
Reconstruct Metamodel using the expanded X_N
End
```

In the above pseudo code $\boldsymbol{D}$ is a vector of length N that contains the minimum of the distances between the current population members and the metamodel construction vectors, $\boldsymbol{l}$ is an index vector matching the indices of $\boldsymbol{D}$ after it has been sorted to the population set $\boldsymbol{P}_{Np}$. *doefac* and *fitfac*, are the fractions of the population considered for inclusion in the metamodel set $\boldsymbol{X_N}$ based on the minimum distance and fitness respectively. η and ε are the minimum distance allowable for inclusion of the points selected based on fitness and distance, respectively.

The metamodel update first starts by sorting the population in a descending order such that $\boldsymbol{p_i}$ is fitter than $\boldsymbol{p_{i+1}}$. The next step involves computation of the minimum distance vector $\boldsymbol{D}$. A fraction of the population (*fitfac*) based on fitness is considered for inclusion in the metamodel. The member under consideration must satisfy a minimum distance check based on η. Similarly, another fraction of the population based on distance criteria is selected. The new expanded metamodel set $\boldsymbol{X_N}$ is finally used for reconstructing the metamodel. Here equation (1) to (8) are used.

This update mechanism does not allow redundant points to be included in the metamodel. The first selection criterion is mathematically equivalent to *maximin* criterion used in design of experiments literature (Torczon and Trosset, 1998). This criterion can lead to space filling designs. In the initial stages of the search this criterion is almost always satisfied by most of the population members, later on in the exploitation phases only very few members satisfy this criterion.

Note that η should always be set greater that ε to allow for proper metamodel fine-tuning during the final exploitative stages of the search.

The process of continuously updating the metamodel creates a dynamic landscape. In such a landscape false optima are likely to appear and disappear. Hence it is likely that the best individual obtained may be lost during the evolution of the metamodel. To prevent this we use elitism. In the first generation the fittest individual is stored. At each of the following generations the best individual is evaluated using the computer model. Its fitness is then compared with that of the stored elite. If the best individual is fitter than the elite, the elite takes on its values, otherwise the best individual is replaced by the stored elite.

4 EXPERIMENTAL RESULTS

We conducted experimental studies on two test functions and one real world problem to evaluate the performance of our approach.

We used a GA for the experiments reported here. It is typical of those described in Michalewicz (1996). 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 50, and the tournament size was set to 2. For the metamodel update parameters *generation delay* was set to 2, *fitfac* set to 0.1 and *doefac* to 0.5.

4.1 THE BUMP FUNCTION

The ‘bump’ problem, introduced by Keane (1994), is very hard for most optimizers to deal with. It is quite smooth but contains many peaks, all of similar heights. Moreover, its optimal value is defined by the presence of a constraint boundary.

The problem is defined as

$$\text{maximize } \frac{\text{abs}\left(\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n}\cos^2(x_i)\right)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \quad (9)$$

for

$$0<x_i<10 \quad i=1,\ldots,n \quad (10)$$

subject to

$$\prod_{i=1}^{n} x_i > 0.75 \quad \text{and} \quad \sum_{i=1}^{n} x_i < 15n/2 \quad (11)$$

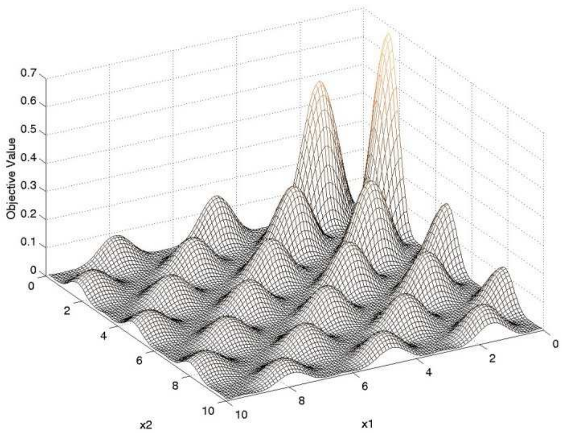

Figure 1: Plot of the bump problem for n=2

An interesting feature of this function is that the surface is nearly, but not quite, symmetrical about x_1=x_2, so the peaks always occur in pairs, but with one always bigger than its sibling. The global optimum is defined by the product constraint. When the problem is generalized for n greater than two, it becomes even more demanding with families of similar peaks occurring within a highly complex constraint surface. These properties of bump have made it suitable for the study of GA performance and optimizing GA control parameters (Keane, 1995a) as well as the control parameters of other evolutionary optimization methods (Keane, 1995b).

For the 2D case η was set to 0.001 and ε to 0.2, and for 5D to 0.0001 and 0.7. m*axeval* was set to 500. A standard GA with identical parameters was setup to see how the metamodeling based strategy compares. The results depicted in Figures 2 and 3 show that a metamodeling strategy is clearly advantageous for this problem. In this problem only the feasible points were used to build the metamodel.

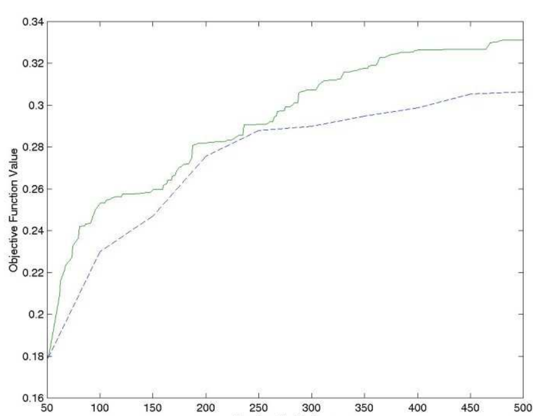

Figure 2: . Best of generation versus number of accurate evaluations averaged over 20 runs for 2D bump. The solid line is for a metamodel based search. The dashed line is for the baseline comparison.

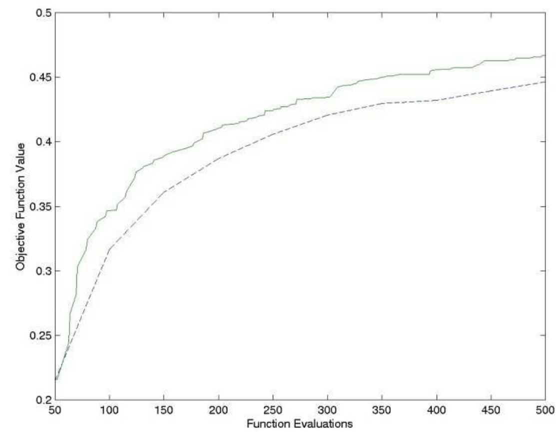

Figure 3: Best of generation versus number of accurate evaluations averaged over 20 runs for 5D bump. The solid line is for a metamodel based search. The dashed line is for the baseline comparison.

For the 2D case, the resulting metamodel after the optimization has been carried out, is shown in Figure 4. It is clear that the metamodel captures the essential feature of bump, especially in the high fitness region.

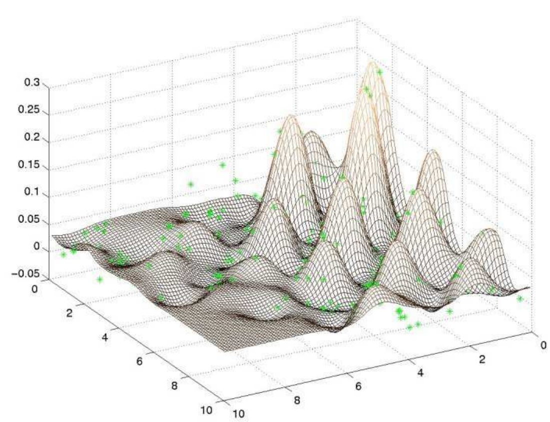

Figure 4: Resultant metamodel after the optimization was carried out. Note the resemblance with Figure 1 for the constrained region.

4.2 GENERALIZED ACKLEY

The generalized Ackley function has one distinct optima but its surface is quite bumpy. The problem is expressed as (Bäck *et a.*, 1993) minimize

$$-a \cdot \exp\left[-b\left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right)\right] - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(x_i^2\right)\right) + a + \exp(1) \tag{12}$$

for

$$-32.768 < x_i < 32.768 \qquad i=1,...,n \tag{13}$$

This function has a know global optima of zero at $x_i=0$ for $i=1,...,n$.

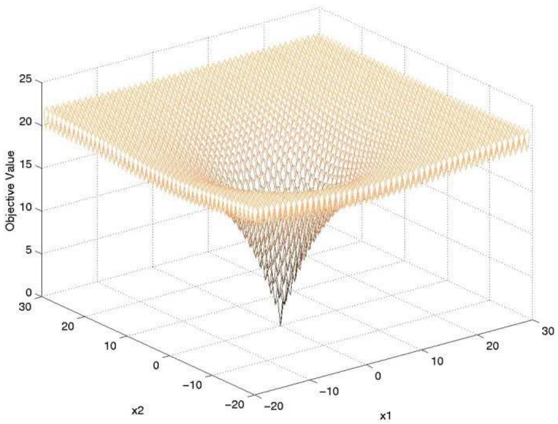

Figure 5: Plot of the Ackley function for n=2.

Here we tried the metamodeling strategy for 2D and 5D cases of Ackley. For the 2D case η was set to 0.0005 and ε to 0.1, and for 5D to 0.00001 and 0.3. *maxeval* was set to 500.

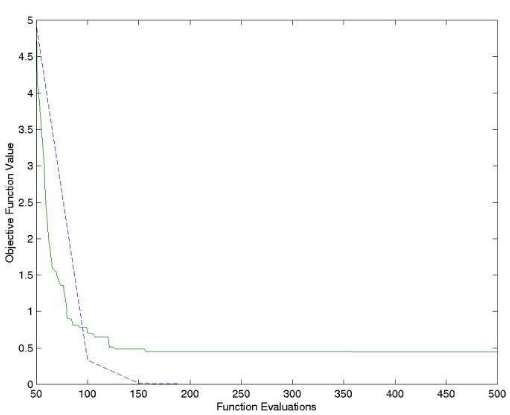

Figure 6: Best of generation versus number of accurate evaluations averaged over 20 runs for 2D Ackley. The solid line is for metamodel based search. The dashed line is for the baseline comparison.

As the above figure clearly shows, the metamodel based search gets near the optima pretty quickly and then stalls. The global DACE metamodel cannot capture the details around the optima while at the same time maintain a good approximation of the other points.

4.3 EXPERIMENTS ON A BEAM STRUCTURE

Further experiments have been carried out on the design problem of a 2D space structure (Keane, 1995c). The baseline structure is shown below in Figure 7. It consists of 40 individual Euler-Bernoulli beams connected at 20 joints. 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 as minimizing the energy level of the end beam at an excitation frequency of 150 Hz. The optimizer was allowed to generate new geometries by varying the coordinates of the inner 18 joints of this structure.

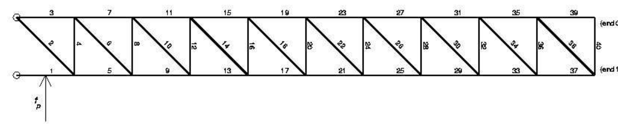

Figure 7: The 2D beam baseline structure

The GA was set to minimize the energy levels in decibels over the baseline structure. For this problem *maxeval* was set to 200.

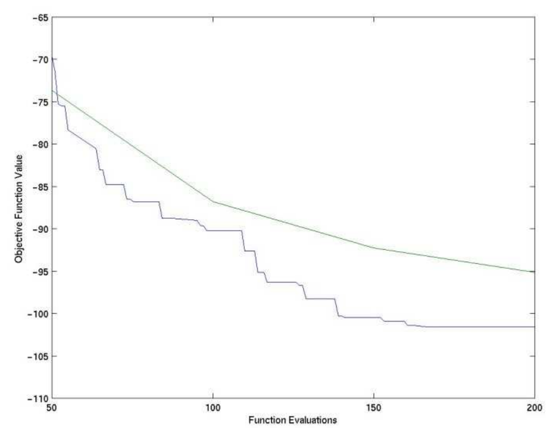

Figure 8: Best of generation versus number of accurate evaluations averaged over 5 runs on the beam structure problem. The solid line is for metamodel based search. The dashed line if for the baseline comparison.

The above graph shows that the metamodeling approach resulted in a superior design.

5 CONCLUSION AND FUTURE WORK

We have presented an algorithm for integrating metamodels with evolutionary optimization procedures. It is shown via experiments on some example problems, that this approach can enable the application of evolutionary algorithms to computationally expensive optimization problems. The metamodeling approach seems to work best with smooth objective functions. It stalls in situations where the global optima has strong local features.

An added advantage of the method is that the resultant metamodel can be used later on for parametric design studies, visualization, etc.

Though metamodels are considered here to incur insignificant computational burden, this will not be the case when many points are added. Metamodeling is essentially a cumulative process and the computational complexity of its construction is of the order N^3. Hence for a large enough N the cost of constructing a metamodel will be quite significant. The use of many local metamodels is expected to alleviate this burden.

Future work will explore the use of a combination of several local metamodels that are tightly coupled to evolutionary processes so as to overcome pathological scenarios and enhance the current performance. Also, the use of local metamodels should enable one to estimate correlation parameters for each subspace independently. This would improve the approximation accuracy for complex multimodal functions. The use of diversity maintenance mechanism in conjunction with the metamodels needs to be further explored.

Acknowledments

This work was supported under EPSRC grant no GR/L04733.

REFERENCES

Bäck T., Rudolph G., and Schwefel H. P. (1993) Evolutionary programming and evolutionary strategies: similarities and differences, in *Proc. 2nd Ann. Conf. on Evolutionary Programming* (Fogel D. and Atmar W., eds), pp. 11-22. Evolutionary Programming Society, Sand Diego, CA.

Booker A. J., Dennis J. E., Frank P. D., Serafini D. B., Torczon V., and Trosset M. W. (1998) A Rigorous Framework for Optimization of Expensive Functions by Surrogates. CRPC-TR98739-S, *Center for Research on Parallel Computation*, Rice University.

Cressie, N. A. C. (1991) *Statistics for Spatial Data*, John Wiley and Sons, Inc., New York.

Dennis J. E. and Torczon V. (1997) Managing approximation models in optimization, in *Multidisciplinary Design Optimization: State of the Art* (Alexandrov N. M. and Hussaini M. Y., eds), pp. 330-347. SIAM, .

Eby D., Averill R. C., Punch W. F., and Goodman E. D. (1998) Evaluation of Injection Island GA Performance on Flywheel Design Optimization, in *Proceedings of Third Conference on Adaptive Computing in Design and Manufacturing* (Parmee I., ed), pp. 121-136. Springer Verlag, London.

El-Beltagy M. A. and Keane A. J. (1998) 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)* (Parmee I., ed), pp. 111-120. Springer, London.

Gibbs M. N. (1997) *Bayesian Gaussian Processes for Regression and Classification*, [Thesis]. University of Cambridge.

Goodman E. D., Averill R. C., Punch W. F., and Eby D. (1997) Parallel Genetic Algorithms in the Optimization of Composite Structures. *Second World Conference on Soft Computing (WSC2).*

Keane, A. J., (1994) Experiences with optimizers in structural design,' in *Proceedings of the Conference on Adaptive Computing in Engineering Design and Control 94*, ed. I. C. Parmee, Plymouth, pp. 14-27.

Keane A. J. (1995a) A brief Comparison of Some Evolutionary Optimization Methods. *Proceedings of the Conference on Applied Decision Technologies (Modern Heuristic Search Methods).*, 125-137.

Keane A. J. (1995b) Genetic Algorithm Optimization of Multi-peak Problems: Studies in Convergence and Robustness. *Artificial Intelligence in Engineering* **9(2)**, 75-83.

Keane, A. J., (1995c), Passive vibration control via unusual geometries: the application of genetic algorithm optimization to structural design, *Journal of Sound and Vibration*, 185(3) pp. 441-453.

Michalewicz Z. (1996) *Genetic Algorithms+Data Structures =Evolution Programs, 3rd ed.* Springer, New York.

Nair P. B., Keane A. J., and Langley R. S. (1999) 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*, pp. AIAA 99-1260. AIAA, Missouri.

Nair P. B., Keane A. J., and Shimpi R. P. (1998) Combining Approximation Concepts With Genetic Algorithm-Based Structural Optimization Procedures. *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures* **2**, 1741-1751.

Neal R. M. (1996) *Bayesian Learning for Neural Networks*, Number 118 in *Lecture Notes in Statistics*, Springer, New York.

Owen, A. B., (1992) Orthogonal Arrays for Computer Experiments, Integration and Visualization,' *Statistica Sinica* 2, pp. 439-452.

Ratle A. (1998) Accelerating the convergence of Evolutionary Algorithms by Fitness Landscape Approximation, in *Parallel Problem Solving from Nature-PPSN V*, .

Robinson G. M. and Keane A. J. (1998) A Case for Multi-level Optimization in Aeronautical Design. Proceedings of the Royal Aeronautical Society Conference on Multidisciplinary Design and Optimization.

Sacks J., Welch W. J., Mitchell T. J., and Wynn H. P. (1989) Design and analysis of Computer experiments. *Statistical Science* **4(4)**, 409-435.

Simpson T. W. (1998) Comparison of Response Surface and Kirging Models in the Multidisciplinary Design of an Aerospike Nozzle. *ICASE Report No. 98-16*, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center.

Simpson T. W., Peplinski J. D., Koch P. N., and Allen J. K. (1997) On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments, in *ASME Design Theory and Methodology* (Shah J., ed), pp. ASME97-DETC97/DTM-3881. ASME, New York.

Torczon, V., and Trosset, M. W., (1998) Using Approximations to Accelerate Engineering Design Optimization," in *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St.Louis, Missouri, September 2-4, AIAA Paper 98-4800.

Vekeria H. D. and Parmee I. (1996) The Use Of A Co-operative Multi-Level CHC GA For Structural Shape Optimisation. Proceedings of the Fourth European *Congress on Intelligent Techniques and Soft Computing*. Aachen, Germany.