International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:10, 2007

Surrogate based Evolutionary Algorithm for Design Optimization

Maumita Bhattacharya

***Abstract*—**Optimization is often a critical issue for most system design problems. Evolutionary Algorithms are population-based, stochastic search techniques, widely used as efficient global optimizers. However, finding optimal solution to complex high dimensional, multimodal problems often require highly computationally expensive function evaluations and hence are practically prohibitive. The Dynamic Approximate Fitness based Hybrid EA (DAFHEA) model presented in our earlier work [14] reduced computation time by controlled use of meta-models to partially replace the actual function evaluation by approximate function evaluation. However, the underlying assumption in DAFHEA is that the training samples for the meta-model are generated from a single uniform model. Situations like model formation involving variable input dimensions and noisy data certainly can not be covered by this assumption. In this paper we present an enhanced version of DAFHEA that incorporates a multiple-model based learning approach for the SVM approximator. DAFHEA-II (the enhanced version of the DAFHEA framework) also overcomes the high computational expense involved with additional clustering requirements of the original DAFHEA framework. The proposed framework has been tested on several benchmark functions and the empirical results illustrate the advantages of the proposed technique.

***Keywords*—**Evolutionary algorithm, Fitness function, Optimization, Meta-model, Stochastic method.

I. Introduction

STOCHASTIC, zero-order search techniques like evolutionary algorithms (EAs) are efficient global optimizers. However, these methods require numerous evaluations of candidate solutions. For many real world problems, like large scale finite element analysis (FEA), such evaluation of one single function could involve hours to days of complete simulation. The solution to this problem lies in reducing the number of such evaluations without significantly compromising the accuracy of the results. Using meta-models of to replace such expensive function evaluation is computationally many orders of magnitude cheaper [8, 10, and 11]. Many regression and interpolation tools (e.g. least square regression, back propagating artificial neural network, response surface models, etc.) can be used for building such surrogate models. However, accuracy of the result is a major risk involved in using meta-models to replace actual function evaluation [22, 24, 25, and 26]. When it is infeasible to precisely judge when, where and how much of such replacement is optimal, a using a controlled approach holds the answer. In our earlier work, we presented a hybrid framework, DAFHEA (a dynamic approximate fitness based hybrid evolutionary algorithm) to address this issue. DAFHEA replaces expensive function evaluation by its support vector machine (SVM) approximation. The concept of *merit function* [21] is borrowed to maintain diversity in the solution space using approximate knowledge. However, the assumption used in DAFHEA is that the training samples for the meta-model are generated from a single uniform model. This does not cover situations, where information from variable input dimensions and noisy data is involved. Our current work attempts to correct this by using a multi-model regression approach. The multiple models are estimated by successive application of the SVM regression algorithm. Retraining of the model is done in a periodic fashion.

The paper is arranged as follows: Section II presents a brief review of some related works. The basic framework and the functional details of DAFHEA-II (the enhanced version of DAFHEA) is presented in Section III and Section IV. Simulation results are presented in Section V. Finally concluding remarks are summarised in Section VI.

Maumita Bhattacharya is with the Charles Sturt University, SEIS, PO Box: 789 , 2640 Albury, NSW, Australia (e-mail: maumita.bhattacharya@ieee.org).

II. Related Works

The use of approximate model to speed up optimisation dates all the way back to the sixties [14]. The most widely used models being Response Surface Methodology [16], Krieging models [9] and artificial neural network models [5]. The concepts of using approximate model vary in levels of approximation (*Problem approximation, Functional approximation, and Evolutionary approximation*), model incorporation mechanism and model management techniques [26].

In the multidisciplinary optimisation (MDO) community, primarily response surface analysis and polynomial fitting techniques are used to build the approximate models [15, 22]. These models work well when single point traditional gradient-based optimisation methods are used. However, they are not well suited for high dimensional multimodal problems as they generally carry out approximation using simple quadratic models.

In another approach, multilevel search strategies are developed using special relationship between the approximate and the actual model. An interesting class of such models focuses on having many islands using low accuracy/cheap evaluation models with small number of finite elements that

progressively propagate individuals to fewer islands using more accurate/expensive evaluations [7]. As is observed in [26], this approach may suffer from lower complexity/cheap islands having false optima whose fitness values are higher than those in the higher complexity/expensive islands. Rasheed et al. in [10, 11], uses a method of maintaining a large sample of points divided into clusters. Least square quadratic approximations are periodically formed of the entire sample as well as the big clusters. Problem of unevaluable points was taken into account as a design aspect. However, it is only logical to accept that true evaluation should be used along with approximation for reliable results in most practical situations. Another approach using population clustering is that of *fitness imitation* [26]. Here, the population is clustered into several groups and true evaluation is done only for the cluster representative [8]. The fitness value of other members of the same cluster is estimated by a distance measure. The method may be too simplistic to be reliable, where the population landscape is a complex, multimodal one.

Jin et al. in [24, 25] analysed the convergence property of approximate fitness based evolutionary algorithm. It has been observed that incorrect convergence can occur due to false optima introduced by approximate model. Two *controlled evolution* strategies have been introduced. In this approach, new solutions (offspring) can be (pre)-evaluated by the model. The (pre)-evaluation can be used to indicate promising solutions. It is not clear however, how to decide on the optimal fraction of the new individuals for which true evaluation should be done [6]. In an alternative approach, the optimum is first searched on the model. The obtained optimum is then evaluated on the objective function and added to the training data of the model [1, 21, and 6]. Yet another approach as proposed in [24], a regularization technique is used to eliminate false minima.

It is obvious that incorporation of approximate models may be one of the most promising approaches to realistically use EA to solve complex real life problems, especially where: (i). Fitness computation is highly time-consuming, (ii). Explicit model for fitness computation is absent, (iii). Environment of the evolutionary algorithm is noisy etc. However, considering the obvious risk involved in such approach, an EA with efficient control strategy for the approximate model and robust performance is welcome.

While dealing with complex real world optimization problems, expensive function evaluations can be feasibly used only in a limited manner. Research on using surrogate models should focus on:

a. Minimizing uncertainty in approximate estimation
b. Employing corrective measures
c. Exploring ways to exploit the approximate knowledge for improving the optimization technique.

The original DAFHEA framework is similar to other models in that it uses an approximation model to partially replace expensive fitness evaluation in evolutionary algorithm. An *explicit* control strategy (*a cluster-based on-line learning technique*) to improve reliability of using such approximate models to reduce expensive function evaluations was introduced. Also the approximate knowledge thus generated is exploited to avoid premature convergence (one of the major impediments of using evolutionary algorithm to solve complex real life optimisation problems). However, the major constraint associated with DAFHEA is that it treats the solution space as one comprising of information coming from a uniform model. Situations like model formation involving variable input dimensions and noisy data certainly can not be covered by this assumption. Our current work addresses this problem by using a multiple model regression approach for the SVM approximator. The following sections explain how this is achieved in the enhanced DAFHEA (DAFHEA-II).

III. The Modified DAFHEA Framework

As in the original DAFHEA framework, DAFHEA-II [Figure 1] includes a global model of genetic algorithm (GA), hybridised with support vector machine (SVM) as the approximation tool. Expensive fitness evaluation of individuals as required in traditional evolutionary algorithm is partially replaced by SVM approximation (regression) models. *Evolution control* is implemented by periodic expensive evaluations, leading to considerable speedup without compromising heavily on solution accuracy. Also the approximate knowledge about the solution space generated is used to maintain population diversity to avoid premature convergence.

While approximation is not a new idea in accelerating iterative optimisation process, DAFHEA-II is specifically suited for applications involving information that could be considered generated by more than one model. As in original DAFHEA, this framework also focuses on controlled speedup to avoid detrimental effects of approximation and exploiting approximate knowledge to improve optimisation solution. The following section presents the basic algorithm structure of DAFHEA-II, while Section IV explains its major functional aspects.

A. Basic Algorithm Structure

The proposed DAFHEA-II framework is introduced in the context of unconstrained optimisation problems. Figure (1) schematically presents the algorithm.

```
/* The basic algorithm for the DAFHEA-II
framework */
Procedure DAFHEA_II
{
  {
  initialize population matrix, gen = 0 and
  set α, β
  call actual function evaluation
  call Procedure train_SVM to generate
  approximation models
  while ( gen < α )
```

```
    {
      gen = gen + 1
      rank solutions based on fitness
      retain actual elite
      apply crossover and mutation to
      generate offspring
      call Procedure predict_SVM to
      approximate the fitness of the
      offspring
      retain approximate elite
      if ( gen mod β = 0 ) then
        {
        call actual function evaluation
        call Procedure train_SVM
        }
      }
      rank  solutions based on fitness
      get the best solution
    }
/* This procedure estimates multiple
models from the training data set */
Procedure train_SVM
{
  initialize data set=population in
  current generation with fitness
  resulting from actual evaluation
  while (!stopping criterion)
  {
/* Estimate the dominant model describing
majority of the candidates in data set */

      apply robust regression to data set
/* Partition the available data */
      {
      analyse available data to separate
      others from majority based on their
      distance from the dominant model
      remove subset of data generated by
      the dominant model from the data set
      }
  }
}
/* This procedure selects the most likely
model for each member of the population to
find its corresponding estimate */

Procedure predict_SVM
{
  while (population member)
  {
  determine appropriate model for the
  given test sample z = (x, y) using a
  distance measure from each model
  }
}
}
```

Fig. 1 The DAFHEA-II Framework

In the above framework (see Figure 1) α is the number of predetermined generations and β is the predetermined retraining frequency.

B. The Implementation of the DAFHEA-II Framework

The detailed implementation of the DAFHEA-II framework is as given below:

Step One: Create a random population of N_c individuals, where, $N_c = 5 * N_a$ and $N_a =$ actual initial population size.

Step Two: Evaluate N_c individual using actual expensive function evaluation. Build the SVM approximate models using the candidate solutions as input and the actual fitness (expensive function evaluation values) as targets forming the training set for *off-line training*. Details of the Multiple Model Formation technique is described in Section IV.

Step Three: Select N_a best individual out of N_c evaluated individuals to form the initial GA population.

Remarks: The idea behind using five times the actual EA population size (as explained in *Step One*) is to make the approximation model sufficiently representative at least initially. Since initial EA population is formed with N_a best individuals out of these N_c individuals, with high recombination and low mutation rates, the EA population in first few generations is unlikely to drift much from its initial locality. Thus it is expected that large number of samples used in building the approximation model will facilitate better performance at this stage. Also using the higher fitness individuals, chosen out of a larger set should give an initial boost to the evolutionary process.

Step Four: Rank the candidate solutions based on their fitness value.

Step Five: Preserve the elite by carrying over the best candidate solution to the next generation.

Step Six: Select parents using suitable selection operator and apply genetic operators namely recombination and mutation to create children (new candidate solutions) for the next generation.

Step Seven: The SVM regression models created in Step two are applied to estimate the fitness of the children (new candidate solutions) created in Step six. This involves assignment of most likely or appropriate models to each candidate solution. The details of this prediction mechanism are given in Section IV.

Step Eight: The set of newly created candidate solutions is ranked based on their approximate fitness values.

Step Nine: The best performing newly created candidate solution and the elite selected in Step five are carried to the population of the next generation.

Step Ten: New candidate solutions or children are created as described in Step six.

Step Eleven: Repeat Step seven to Step ten until either of the following condition is reached:

i. The predetermined maximum number of generations has been reached; or

ii. The periodic retraining of the SVM regression models is due.

Step Twelve: If the periodic retraining of the SVM regression models is due, this will involve actual evaluation of the candidate solutions in the current population. Based on this training data new regression models are formed. The algorithm then proceeds to execute Step four to Step eleven.

Remarks: The idea behind using periodic retraining of the SVM regression models is to ensure that the models continue to be representatives of the progressive search areas in the solution space.

IV. Major Functional Aspects of DAFHEA-II

The following sections detail the major functional aspects of the DAFHEA-II framework.

A. Single Approximation Model Formation Using SVM Regression

The theoretical background of support vector machine is mainly inspired from statistical learning theory [23]. Major advantages of the support vector machines over other machine learning models such as neural networks, are that there is no local minima during learning and the generalization error does not depend on the dimension of the space. Also the fast learning ability of the SVM regression [4, 2] model is a desirable property for on-line learning. In DAFHEA (both original and enhanced versions), as the approximation model has to be rebuilt frequently to be representative of the progressing solution space, this is an important criterion for model selection.

Let us consider the problem of approximating the set of data,

$$D = \{(x^1, y^1),...,(x^l, y^l)\} \tag{1}$$

with a linear function,

$$f(x) = w.x + b, \quad w, x \in R^n, b \in R \tag{2}$$

The construction of a model is reduced to the minimization of the following regularized ε -insensitive loss function:

$$L = \|w\|^2 + C.\frac{1}{l}\sum_{i=1}^{l} \max\{|y_i - f(x_i)| - \varepsilon\} \tag{3}$$

where ε is the tolerable error, C is a pre-specified regularization constant and f is the function to be estimated.

The minimization of (3) is equivalent to the following constrained optimisation problem, giving the optimal regression function as:

$$\textbf{min} \quad \frac{1}{2}\|w\|^2 + C \cdot \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{4}$$

$$\textit{subject to} \quad ((w \cdot x_i) + b) - y_i \le \varepsilon + \xi_i \tag{5}$$

$$y_i - ((w.x_i) + b) \le \varepsilon + \xi_i^* \tag{6}$$

$$\xi_i, \xi_i^* \ge 0, \quad i = 1,...,l \tag{7}$$

where ξ_i and ξ_i^* are slack variables representing upper and lower constraints on the output of the system.

Thus, quadratic-programming techniques can be applied to solve the minimization problem.

In the enhanced version of DAFHEA (DAFHEA-II), a multiple model regression approach is used. The technique used closely follows the approach described in [20].

B. Multiple Model Regression

The multiple model regression involves the following two stages:

i. The *training/learning phase*, involving creation of the models based on training data.

ii. The *prediction phase*, involving assignment of the most likely model to each candidate data and estimation of improved response using the selected model.

1) The Training/Learning Phase

Let us consider a finite number of samples or training data $(\mathbf{x}_i, y_i), (i = 1,...,n)$ [20]. The learning involves two objectives:

(a) to estimate N target models from a set of possible models:

$$f_m(\mathbf{x}, \omega_m), (\omega_m \in \mathrm{E}_m, m = 1,...N) \tag{8}$$

Where E_m is a parametric space for model *m*. Each model estimate approximates the corresponding target model $f_m(\mathbf{x}, \omega_m^*) \to tr_m(\mathbf{x})$.

(b) to partition available training data set into N subsets, where each subset belong to an appropriate model. The input (**x**) and/or output (y) space will be thus partitioned into N disjoint regions.

It is clear from the above discussion that the creation of multiple models here can be viewed as a generalization of the traditional single-model estimation. Traditional regression is applied to estimate appropriate regression-like models in a progressive manner while partitioning the data set into subsets at the same time.

2) The Prediction Phase

Using a single model approach, estimating a response $\hat{y}$ for a given test input $\mathbf{x}$, simply amounts to deducing $\hat{y} = f(\mathbf{x}, \omega^*)$, where $f(\mathbf{x}, \omega^*)$ is a model predetermined based on the training data. In case of multiple model estimation, first an appropriate model has to be selected for the test input $\mathbf{x}$ and then the response $\hat{y}$ can be computed as $\hat{y} = f_c(\mathbf{x}, \omega^*)$, where $f_c(\mathbf{x}, \omega^*)$ is the specifically chosen model for $\mathbf{x}$. However, it is not possible to select a model using $\mathbf{x}$ alone, as there may be overlapping of input domains for different models. Thus, selection of model should be based on the $(\mathbf{x}, y)$ values of the test data as described in

[20]. For a given sample of test data $z = (\mathbf{x}, y)$ generated by an unknown model u and a set of models estimated during training stage:

$$f_i(\mathbf{x}, \omega_i^*), \mathbf{x} \in X_i (i = 1,...N) \quad (9)$$

to determine the appropriate or most likely model the *distance* between the test sample and each of the models in (9), has to be computed. Each model in (9) is defined as a region in the input (x) space and the mapping $f_c : \mathbf{x} \to y$ in this region. Therefore, the *distance* may be defined in the input (x) space or in the y-space, or some combination of the two.

3) Exploiting Approximate Knowledge to Avoid Premature Convergence

The basic idea is to encourage *exploration* along with *exploitation* in the initial generations. The underlying notion is to achieve this by attaching a *merit point* to:

i. An individual that belongs to a subset that is furthest from the set that is associated with the dominant model;
ii. An individual coming from a sparsely populated, i.e., inadequately represented subset.

The *merit function* $f_m(x)$ is conceptually similar to the one used in our earlier work [14].

V. Experiment Results And Discussions

The performance of the proposed algorithm is tested on five popular benchmark test functions (see Table 1): namely, Spherical, Rosenbrock, Rastrigin, Schwefel and Ellipsoidal. These benchmark functions in the test suit are scalable and are commonly used to assess the performance of optimization algorithms [12]. For all five functions except Rosenbrock the global minimum is $f(x) = 0$ at $\{x_i\}^n = 0$. Rosenbrock has a global minimum of $f(x) = 0$ at $\{x_i\}^n = 1$.

All simulations were carried out using the following assumptions: The population size of $10n$ was used for all the simulations, where n is the number of variables for the problem; for comparison purposes three sets of input dimensions are considered; namely, $n = 5, 10$ and 20. For all three cases, tenfold validation was done with the number of generations being 1000; the SVM regression models were trained with *five times* the real GA population size initially and all the simulation processes were executed using a Pentium® 4, 2.4GHz CPU processor. Tables II, III and IV show the comparative statistics of the various simulations runs using canonical GA model which uses only actual function evaluations and the proposed DAFHEA-II model which uses actual function evaluations sparingly. We report the results for the 5-D (dimension), 10-D (dimension) and 20-D (dimension) scenarios. The reported results were obtained by achieving same level of tolerance for both canonical GA and the proposed model. For comparison purpose, results reported in [12] were considered (see Table II, III and IV).

TABLE I
List of Test Functions

Function	Formula
Spherical	$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$
Ellipsoidal	$f(\mathbf{x}) = \sum_{i=1}^{n} i x_i^2$
Schwefel	$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{n-1} (x_i - 1)^2 + 100(x_i^2 - x_{i+1})^2$
Rastrigin	$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$

TABLE II
Total Function Evaluations Required (5-D)

Function	Canonical GA (Actual function evaluations)	Proposed DAFHEA-II Framework (Actual function evaluations)
Spherical	49045	21200
Ellipsoidal	49045	21000
Schwefel	49045	25500
Rosenbrock	18000	7009
Rastrigin	16500	4545

TABLE III
Total Function Evaluations Required (10-D)

Function	Canonical GA (Actual function evaluations)	Proposed DAFHEA-II Framework (Actual function evaluations)
Spherical	99150	77500
Ellipsoidal	99150	84300
Schwefel	99150	53750
Rosenbrock	16500	6985
Rastrigin	17100	7175

TABLE IV
Total Function Evaluations Required (20-D)

Function	Canonical GA (Actual function evaluations)	Proposed DAFHEA-II Framework (Actual function evaluations)
Spherical	199200	110400
Ellipsoidal	199200	81450
Schwefel	199200	144200
Rosenbrock	70447	21150
Rastrigin	101650	28000

It is clear from the depicted results that the proposed DAFHEA-II model effectively reduces the number of actual

evaluations for all the benchmark function in our test suit. It is true that the formation and maintenance of the regression models incorporates additional computational expense. However, this approximation based evolutionary algorithm model is not proposed for regular optimisation problems where actual function evaluation is not a matter of concern. Complex real world problems involving very expensive function evaluations will benefit from such approximation based algorithms even when the reduction in the number of actual evaluations is relatively low.

VI. Conclusions And Future Work

Improvement in terms of number of function evaluations to reach an optimum or a near optimum is a topical issue in EA research. This can drastically lower the computational expense of using EA to solve complex design optimisation problems. In this research, a multiple model approach for support vector machine regression was used to perform knowledge extraction. The algorithm showed reliable performance in terms of accuracy and the overhead cost towards developing and maintaining the meta-model is not alarmingly high. Since this overhead is expected not to increase much with increased problem complexity, DAFHEA-II should lead to considerable speed up for complex real life problems. The DAFHEA-II framework is an enhancement over our earlier work [14]. The new framework is specifically designed to solve complex real world optimisation problems where the input information is expected to be generated by multiple models instead of a single mpdel. Our future research will investigate the possibilities of reducing the overhead incurred by formulating and maintaining the regression models.

Acknowledgement

This work was partly supported by a Seed Grant funded by Charles Sturt University, Australia.

References

[1] A. Ratle.., "*Accelerating the convergence of evolutionary algorithms by fitness landscape approximation*", Parallel Problem Solving from Nature-PPSN V, Springer-Verlag, pp. 87-96, 1998.

[2] A. Smola and B. Schölkopf, "*A Tutorial on Support Vector Regression*", NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.

[3] B. Dunham, D. Fridshal., R. Fridshal and J. North, "*Design by natural selection*", Synthese, 15, pp. 254-259, 1963.

[4] B. Schölkopf , J. Burges and A. Smola, ed., "*Advances in Kernel Methods: Support Vector Machines*", MIT Press, 1999.

[5] C. Bishop, "*Neural Networks for Pattern Recognition*", Oxford Press, 1995.

[6] D. Büche., N. Schraudolph, and P. Koumoutsakos, "*Accelerating Evolutionary Algorithms Using Fitness Function Models*", Proc. Workshops Genetic and Evolutionary Computation Conference, Chicago, 2003.

[7] H. D. Vekeria and i. C. Parmee, "*The use of a co-operative multi-level CHC GA for structural shape optimization*", Fourth European Congress on Intelligent Techniques and Soft Computing – EUFIT'96, 1996.

[8] H. S. Kim and S. B. Cho, " *An efficient genetic algorithm with less fitness evaluation by clustering*", Proceedings of IEEE Congress on Evolutionary Computation, pp. 887-894, 2001.

[9] J. Sacks, W. Welch, T. Mitchell and H. Wynn, "*Design and analysis of computer experiments*", Statistical Science, 4(4), 1989.

[10] K. Rasheed, "*An Incremental-Approximate-Clustering Approach for Developing Dynamic Reduced Models for Design Optimization*", Proceedings of IEEE Congress on Evolutionary Computation, 2000.

[11] K. Rasheed, S. Vattam and X. Ni., "*Comparison of Methods for Using Reduced Models to Speed Up Design Optimization*", The Genetic and Evolutionary Computation Conference (GECCO'2002), 2002.

[12] K. Won, T. Roy and K. Tai, "*A Framework for Optimization Using Approximate Functions*", Proceedings of the IEEE Congress on Evolutionary Computation' 2003, Vol.3, IEEE Catalogue No. 03TH8674C, ISBN 0-7803-7805-9.

[13] M. A. El-Beltagy and A. J. Keane, "*Evolutionary optimization for computationally expensive problems using Gaussian processes*", Proc. Int. Conf. on Artificial Intelligence (IC-AI'2001), CSREA Press, Las Vegas, pp. 708-714, 2001.

[14] M. Bhattacharya and G. Lu, "*DAFHEA: A Dynamic Approximate Fitness based Hybrid Evolutionary Algorithm*", Proceedings of the IEEE Congress on Evolutionary Computation' 2003, Vol.3, IEEE Catalogue No. 03TH8674C, ISBN 0-7803-7805-9, pp. 1879-1886.

[15] P. Hajela and A. Lee., "*Topological optimization of rotorcraft subfloor structures for crashworthiness considerations*", Computers and Structures, vol.64, pp. 65-76, 1997.

[16] R. Myers and D. Montgomery, "*Response Surface Methodology*", John Wiley & Sons, 1985.

[17] S. Pierret, "*Three-dimensional blade design by means of an artificial neural network and Navier-Stokes solver*", Proceedings of Fifth Conference on Parallel Problem Solving from Nature, Amsterdam, 1999.

[18] S. R. Gunn, "*Support Vector Machines for Classification and Regression*", Technical Report, School of Electronics and Computer Science, University of Southampton, (Southampton, U.K.), 1998.

[19] T. Hastie, R. Tibshirani, J. Friedman, "*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*", Springer Series in Statistics, ISBN 0-387-95284-5.

[20] V. Cherkassky and Y. Ma, "Multiple Model Estimation: A New Formulation for Predictive Learning", under review in IEE Transaction on Neural Network.

[21] V. Torczon and M. W. Trosset, "*Using approximations to accelerate engineering design optimisation*", ICASE Report No. 98-33. Technical report, NASA Langley Research Center Hampton, VA 23681-2199, 1998.

[22] V. V. Toropov, a. A. Filatov and A. A. Polykin, "*Multiparameter structural optimization using FEM and multipoint explicit approximations*", Structural Optimization, vol. 6, pp. 7-14, 1993.

[23] V. Vapnik, "*The Nature of Statistical Learning Theory*", Springer-Verlag, NY, USA, 1999.

[24] Y. Jin, M. Olhofer and B. Sendhoff, "*A Framework for Evolutionary Optimization with Approximate Fitness Functions*", IEEE Transactions on Evolutionary Computation, 6(5), pp. 481-494, (ISSN: 1089-778X). 2002.

[25] Y. Jin, M. Olhofer and B. Sendhoff., "*On Evolutionary Optimisation with Approximate Fitness Functions*", Proceedings of the Genetic and Evolutionary Computation Conference GECCO, Las Vegas, Nevada, USA. pp. 786- 793, July 10-12, 2000.

[26] Y. Jin., "*A comprehensive survey of fitness approximation in evolutionary computation*", Soft Computing Journal, 2003 (in press).