DAFHEA: A Dynamic Approximate Fitness-based Hybrid EA for Optimisation Problems

Maumita Bhattacharya and Guojun Lu
Gippsland School of Computing and Information Technology,
Monash University,
Victoria, Australia-3842
{Maumita.Bhattacharya, Guojun.Lu}@infotech.monash.edu.au

Abstract - A dynamic approximate fitness-based hybrid evolutionary algorithm is presented here. The proposed model partially replaces expensive fitness evaluation by an approximate model. A cluster-based intelligent guided technique is used to decide on use of expensive function evaluation and dynamically adapt the predicted model. Avoiding expensive function evaluations speeds up the optimisation process. Also additional information derived from the predicted model at lower computational expense, is exploited to improve solution. Experimental findings support the theoretical basis of the proposed framework.

1 Introduction

While evolutionary algorithms (EA) have been long accepted as efficient global optimisers, finding optimal solution to complex high dimensional, multimodal problems often require very expensive fitness function evaluations. In real world problems, one single function evaluation could involve hours to days of complete simulation! Examples of such problems include large-scale finite element analysis (FEA), computational fluid dynamics (CFD), engineering design problems etc. Needless to say, this could brand any iterative optimisation technique to be the most crippling choice to handle such problems. Approximate models of the expensive function evaluation codes are found order of magnitude cheaper computationally [16, 4, 13]. While many regression and interpolation tools (e.g. least square regression, back propagating artificial neural network, response surface models, etc.) can be used for building the approximate model, there are obvious risk factors involved in replacing true values by approximate models. It is often hard to build approximate models that converge to the true optimum or near-true optimum. For example, unguided approximation can mislead the search process to irrevocable end by introducing high levels of noise or false optima. Keeping the overhead of developing and maintaining such approximate model reasonably low is another major challenge.

In this paper we present DAFHEA: a dynamic approximate fitness based hybrid evolutionary algorithm. The basic idea behind the proposed framework is to:

- Use guided approximation to speed up search process.
- Exploit approximate information to avoid premature convergence.

DAFHEA partially replaces expensive function evaluation by its approximate model. The approximation model is realised with a support vector machine (SVM) [22] regression model [5]. Dynamic updating of the approximate model is achieved by adding information on unrepresented regions in the training set, as the search progresses in the solution space, using a cluster-based approach. The concept of merit function [20] is borrowed to maintain diversity in the solution space using approximate knowledge.

The paper is organised as follows: Section 2 presents a brief review of some related works. The basic framework of DAFHEA is presented in Section 3. Section 4 provides theoretical background of various functional aspects of DAFHEA. In Section 5 we have presented some supporting simulation results and discussion thereof. Finally conclusions and future research directions are summarised in Section 6.

2 Related Works

The use of approximate model to speed up optimisation dates all the way back to the sixties [3]. The most widely used models being Response Surface Methodology [12], Krieging models [17] and artificial neural network models [1]. The concept of using approximate model vary in levels of approximation (Problem approximation, Functional approximation, Evolutionary approximation), model incorporation mechanism and model management techniques [10].

In the multidisciplinary optimisation (MDO) community, primarily response surface analysis and polynomial fitting techniques are used to build the approximate models [6, 21]. 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 focus 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 [23]. As is observed in [10], 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 [14, 15], 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 unvaluable 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 [10]. Here, the population is clustered into several groups and true evaluation is done only for the cluster representative [11]. 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 [8, 9] 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 [2]. 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 [16, 20, 2]. Yet another approach as proposed in [9], 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.

The proposed DAFHEA framework is similar to other models in that it uses an approximation model to partially replace expensive fitness evaluation in evolutionary algorithm. Unlike the existing frameworks, DAFHEA uses an explicit control strategy (a cluster-based on-line learning technique) to improve reliability of using such approximate models to reduce expensive function evaluations. 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). The following sections explain how these objectives are achieved in DAFHEA.

3 The DAFHEA Framework

The DAFHEA framework [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 a SVM approximation (regression) model. Explicit control strategies are used for evolution control, 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.

![Figure 1. The DAFHEA Framework](image)

While approximation is not a new idea in accelerating iterative optimisation process, DAFHEA focuses on controlled speedup to avoid detrimental effects of approximation and also exploiting approximate knowledge to improve optimisation solution. The following section presents the basic algorithm structure of DAFHEA, while Section 4 explains its major functional aspects.

3.1 Basic Algorithm Structure

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

**Step Two:** Evaluate $N_c$ individual using actual expensive function evaluation. Build the SVM approximate model using normalised expensive function evaluation values as training set for off-line training. Use of normalised values in the training set appears to improve performance of meta
model, reducing effects of unnaturally high or low values). SVM hyper-parameters are initially tuned based on this training set.

**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:** Select parents using suitable selection operator and apply genetic operators namely recombination and mutation to create new generation.

**Step Five:** Use SVM approximation model to compute fitness of new generation individuals based on approximate evaluation. Form $m$ distance-based (considering spatial distribution of individuals) clusters in the new population space. If for some $n$ clusters, the standard deviation $\sigma \geq$ Predefined Threshold, rearrange solution space into $m+n$ clusters. Compute a merit function $f_m(x)$ as below:

$$f_m(x) = f_a(x) - \rho_1 \sigma_i - \rho_2 d_{ij} - \rho_3 s_i$$

where, $f_a(x)$ is the predicted fitness function value. $\sigma_i$ is standard deviation (in terms of objective value) for the $i^{th}$ cluster and $d_{ij}$ is the normalized minimum Euclidean distance of $j^{th}$ point of $i^{th}$ cluster from all truly evaluated points so far [20]. $s_i$ is the sparseness of the $i^{th}$ cluster. $\rho_1$, $\rho_2$ and $\rho_3$ are scaling factors for $\sigma_i$, $d_{ij}$ and $s_i$ respectively.

$$s_i = \frac{\text{No of individuals in cluster } i}{\text{Dimension of individual}}$$

Further details on the merit function are given in Section 4.3.

**Step Six:** Dynamically update the approximate model as below:

i. Identify the cluster containing the optimum based on approximation.

ii. Perform expensive evaluation for the approximate optimum and its $k$ nearest neighbours. Also perform expensive evaluation for the centroid of all other data clusters and their $k$ nearest neighbours. Expand neighbourhood for true evaluation until a point is found in each space dimension such that % error $\delta \leq$ Predefined threshold. Here,

$$\delta = \frac{a_n - a_p}{a_p} \times 100$$

where, $a_n =$True value of the $i^{th}$ neighbour and $a_p =$Predicted value of the $i^{th}$ neighbour and $\max i = k$.

iii. Add the newly evaluated points to approximate model training set to update model.

**Step Seven:** When termination/evolution control criteria (the evolution control strategy used is as explained in Section 4.2) are not met, repeat Step Four to Step Seven.

It must be noted, the optimum is considered based on the original predicted value $f_a(x)$. For all other purposes fitness based on the merit function $f_m(x)$ is considered.

Periodic parameter tuning of the SVM approximation model was incorporated, though no specific criterion was used.

## 4 Major Functional Aspects of DAFHEA

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

### 4.1 Approximation Model Formation with SVM Regression

The theoretical background of support vector machine is mainly inspired from statistical learning theory [22]. 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 [18, 19] model is a desirable property for online learning. In DAFEA, 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,

$$D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

with a linear function,

$$f(x) = w.x + b, \ w, x \in R^n, b \in R$$

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

$$L = \|w\|^2 + C \sum_{i=1}^{n} \max \{0, |y_i - f(x_i)| - \varepsilon\}$$

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

The minimization of (6) is equivalent to the following constrained optimisation problem, giving the optimal regression function as:

$$f(x) = w.x + b$$
\[
\min \frac{1}{2} \|\mathbf{v}\| + C \cdot \frac{1}{2} \sum_{i=1}^{l} (\xi_i + \xi_i^*)
\]

subject to \((w \cdot x_i + b) - y_i \leq \varepsilon + \xi_i\)

\[
y_i - (w \cdot x_i + b) \leq \varepsilon + \xi_i^*
\]

where \(\xi_i\) and \(\xi_i^*\) 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.

The approximation model is built, using data obtained by performing expensive evaluation on \(N_c\) randomly generated individuals. Initial tuning of the hyper parameters is done based on this training set. The model is dynamically updated using cluster-based on-line learning technique as discussed in Section 4.2. Further tuning of parameters is done every specific number of generations. The approximate model is updated in each generation as described in the following section. Normalisation of target values of training set was used to ensure better numerical stability of the approximation model.

### 4.2 Control Strategy for Approximation Model

The proposed DAFHEA framework uses a cluster-based on-line learning technique with the following objectives:

i. Improve overall prediction quality of the approximate model, encouraging exploration along with exploitation in the initial stages.

ii. Adapt the model to be more and more representative of the current decision surface in later generations.

#### Cluster-based on-line learning strategy

A hierarchical control strategy (Figure 2) is used as below:

**Level 1:** As is mentioned in Section 3.1, in each generation the population space is divided into dynamic clusters using a simple K-means clustering algorithm, as below:

The dissimilarity between individual \(i\) and \(i'\) is given by their squared Euclidean distance:

\[
d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|x_i - x_{i'}\| ^2
\]

where \(p\) is the dimension of each individual. The within-point-scatter loss function can be defined as:

\[
W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(j) = k} \sum_{j \in C(j)} \|x_j - \bar{x}_k\|^2 = \sum_{k=1}^{K} \sum_{i \in C(j) = k} \|x_i - \bar{x}_k\|^2
\]

where \(\bar{x}_k = (\bar{x}_{1k}, ..., \bar{x}_{pk})\) is the mean vector associated with the \(k^{th}\) cluster. The clustering process involves minimizing the criterion by assigning the \(N\) observations to the \(K\) clusters in a way such that within each cluster the average dissimilarity of the observations from the cluster mean, as defined by the points in that cluster is minimized [7]. The problem reduces to solving the following:

\[
\min \sum_{k=1}^{K} \sum_{i \in C(j) = k} \|x_i - \bar{x}_k\|^2
\]

The K-means clustering algorithm has been chosen so that the population space is grouped into distance-based clusters with less regard to cluster size. As mentioned in Section 3.1, for some \(n\) clusters out of a total of \(m\) clusters, if the standard deviation \(\sigma \geq Predefined\ Threshold\), rearrange solution space into \(m+n\) clusters. This re-clustering makes the regions/clusters more representative of the specific parts of the population landscape.

Now, expensive evaluation is done for the approximate optimum and its \(k\) – nearest neighbours, starting with a high value of \(k\), as below:

For a given point \(x_0\), we find the \(k\) neighbouring points \(x(r)\), \(r = 1, ..., k\) closest to Euclidean distance \(d(r) = \|x(r) - x_0\|\) to \(x_0\). The expensive evaluation is performed for \(x_0\) and \(x(r), r \in [1, k]\) until for each space dimension \(i, i \in [1, p]\), we have a point whose approximation error \(\delta_i \leq Predefined\ threshold\, or\ the\ limit\ \(k\) is reached.

![Figure 2. Hierarchical control strategy used in DAFHEA](image-url)

Using the above technique, expensive evaluation is also done for all the cluster centroids and their \(k\) – nearest neighbours in the initial EA generations.

Repetition of expensive evaluation of the points is avoided using an archive of the already evaluated points. All true evaluation values are added to the training set of the approximation model to rebuild it. Replacing the least recently considered points from the training set by new entries, controls the undefined growth of the training set size.
Level 2: In later stages, the approximate model training set contains information pertaining to the best individual of the previous generation, \( x_{\text{best}} \) with its \( k \) nearest neighbors and \( N \) most recently evaluated solutions. This is intended to make the approximate model more and more representative of the current decision space.

Level 3: Stage 2 continues until eventually true evaluation completely replaces the approximate model.

Remarks: True evaluation of the approximate optimum \( x_{\text{best}} \) helps to identify the presence of false minima, if any. False minima occur mainly due to absence of representative training data in that region. Introducing these true values in the approximate model training set corrects this problem. To further improve prediction in that region, expensive evaluation is continued in the nearest neighbourhood of \( x_{\text{best}} \), until satisfactory approximation is observed within predefined range or limit for number of neighbours \( k \) is reached. Repeating this exercise for other cluster centroids, improves prediction quality of the model in the sparsely explored regions of the solution space, identifying regions where adequate training sample are unavailable. Level 2 and Level 3 (as mentioned above) ensures that the approximate model focuses on the current search space and better accuracy is achieved in the final stage.

4.3 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 merit point to:

i. An individual that belongs to a region with high variance of fitness values;

ii. An individual that is farthest from all the truly evaluated points so far, i.e., belongs to an inadequately explored region;

iii. An individual coming from a sparsely populated, i.e., inadequately represented region.

The merit function \( f_m(x) \) (as defined in Equation 1 in Section 3.1) is introduced to ensure a better spread of data points in the search space. Significance of the three merit factors \( \sigma_i \), \( d_j \) and \( s_l \) is as follows:

\( \sigma_i \) is the standard deviation (in terms of objective value) of cluster \( i \). High \( \sigma_i \) denotes presence of peaks and valleys in the region and attaching a merit point encourages more rigorous search. \( d_j \) is expressed as \( \min \left\| x - x_j \right\|_2 \) and is computed over all \( x_j \) whose true objective value is known [20]. Thus the point \( x_j \) which is farthest from the already considered search site, gets the maximum bonus. We also consider a sparseness measure \( s_l \) (as defined in Equation 2 in Section 3.1). Use of \( s_l \) as a merit factor encourages search in thinly populated regions. However, the scaling factors \( \rho_1 \), \( \rho_2 \) and \( \rho_3 \) should be set carefully to represent relative importance of \( \sigma_i \), \( d_j \) and \( s_l \). The merit function \( f_m(x) \) is used only in the initial generations, along with cluster-based on-line data sampling. While this approach does not guarantee avoidance of premature convergence, it is certainly most likely to encourage maintenance of diversity in the decision space.

5 Experiment Results and Discussions

The performance of the proposed algorithm is analysed on two popular benchmark test functions, namely:

i. The Rosenbrock’s valley or the Banana function defined as:

\[
f(x) = \sum_{i=1}^{n-1} \left[ 100(x_i^2 - x_{i+1})^2 + (1 - x_{i+1})^2 \right]
\]

where, \(-2.048 \leq x_i \leq 2.048\)

ii. De Jong’s function 1 defined as:

\[
f(x) = \sum_{i=1}^{n} x_i^2 \]

where, \(-5.12 \leq x_i \leq 5.12\)


Figure 3a. True surface (Rosenbrock’s valley function or the Banana function)

The SVM regression model was trained with 500 (five times the real GA population size of 100. Reasons are as explained in Section 3.1) randomly generated samples in both test cases. This was done on purpose to better simulate practical scenarios with limited training samples. Alternatively, training samples collected from a few consecutive real EA generations would be more representative. The initial GA population however is formed with 100 best individuals from the training set of 500 individuals. Figure 3a, 3b, 3c and Figure 4a, 4b, 4c show the surfaces of the considered functions in the following three cases:

i. True function landscape;

ii. Uncontrolled pure SVM Approximation;

iii. Controlled (as explained in Section 4.2) SVM approximation with DAFHEA.

The uncontrolled and controlled scenarios shown in the figures are considered after same number of generations, to realise the effect of the control strategy. As can be seen, the proposed cluster-based on-line learning of the
approximation model, improves performance in both cases. Convergence of DAFHEA as against canonical GA with true evaluation is depicted in Figure 5 and Figure 6. For both test functions, the proposed model converges with reasonable degree of accuracy. Number of true function evaluations could be further improved compromising on accuracy. Training time for SVM regression model was in order of tens of seconds in all cases. Even with increased complexity, the overhead towards maintaining the approximation model should be reasonably low. As explained in Section 4.2, the growth of the training set size (for SVM approximation model) is checked by replacing least recently considered points by new entries, thus controlling the increase in training time due to increased training set size.

Figure 3b. SVM approximation model learnt surface (Rosenbrock’s valley function)

Figure 3c. Controlled SVM approximation model learnt surface (Rosenbrock’s valley function)

6 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. Various learning methods can be used to extract convergence knowledge from the current set of truly evaluated solutions to build low accuracy/cheaper models.

In this research, a support vector machine regression model was used to perform knowledge extraction. A cluster-based on-line learning strategy helped the approximate model to move with the EA search process. Efficient learning is achieved by identifying representative unlearned regions of the search space as the search progresses. The idea was to eliminate the uncertainty involved with other random/ blind approaches to train the approximate model. Also the approximate knowledge thus generated was further exploited to avoid premature convergence by encouraging maintenance of diversity in the decision space. A carefully designed merit function achieved this goal. Experimental results showed that the proposed technique is capable of optimising functions of varied complexity (highly correlated and mis-scaled Rosenbrock’s function and the quadratic De Jong function 1) efficiently. 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 should lead to considerable speed up for complex real life problems.

However, certain issues demand future attention. DAFHEA’s dependence on a number of pre-defined parameters (as in Equation 1), thresholds should be carefully investigated to achieve robust performance. The K-means clustering algorithm was used to divide the solution space into representative sub-groups. Other clustering techniques like fuzzy C-means algorithm (FCM), self-organizing map (SOM) etc. will be tried to identify more representative search regions. Also better strategy to eliminate false minima will be investigated. A prospective approach is to incorporate an uncertainty measure of the approximation model, for reliable modelling. Approximate fitness function has been so far mainly used in the cases, where the computation is time-consuming. No explicit model for fitness computation is available for noisy environment or highly multimodal landscapes. Future research will explore these areas. Finally investigating the use of parallelization to further speed up the evolutionary process is definitely worthwhile.

Bibliography


