Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization

Z. Zhou[*], Y. S. Ong[*], P. B. Nair[†], A. J. Keane[†], K. Y. Lum[‡]

Abstract

In this paper, we present a novel surrogate-assisted evolutionary optimization framework for solving computationally expensive problems. The proposed framework uses computationally cheap hierarchical surrogate models constructed through online learning to replace the exact computationally expensive objective function during evolutionary search. At the first level, the framework employs a data parallel Gaussian Process based global surrogate model to filter the EA population of promising individuals. Subsequently, these potential individuals undergo a memetic search in the form of Lamarckian learning at the second level. The Lamarckian evolution involves a trust-region enabled gradient-based search strategy that employs radial basis function local surrogate models to accelerate convergence. Numerical results are presented on a series of benchmark test functions and an aerodynamic airfoil design problem. The results obtained suggest that the proposed optimization framework converges to good designs on a limited computational budget. Furthermore, it is shown that the new algorithm gives significant savings in computational cost when compared to the traditional evolutionary algorithm and other surrogate assisted optimization frameworks.

Index Terms

Evolutionary Optimization, Global and Local Surrogate Model, Genetic Algorithm, Gaussian Process, Radial Basis Function, Aerodynamic Shape Design

Manuscript received September 30, 2004.

Corresponding Author: Y. S. Ong, Contact no: +65-6790-6448; fax: +65-6792-6559; e-mail: asysong@ntu.edu.sg

[*] Z. Zhou and Y. S. Ong are with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798.(e-mail: zhou0018@ntu.edu.sg, asysong@ntu.edu.sg).

[†] P. B. Nair and A. J. Keane are with the Computational Engineering and Design Group, School of Engineering Sciences, University of Southampton, Highfield, Southampton SO17 1BJ, England (e-mail: pbn@soton.ac.uk, ajk@soton.ac.uk).

[‡] K. Y. Lum is with the Temasek Laboratories, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260 (e-mail: kaiyew_lum@nus.edu.sg).

I. Introduction

Evolutionary Algorithms (EAs) have been successfully applied to many complex engineering design optimization problems in recent years. Their popularity lies in ease of implementation and their ability to converge close to the global optimal design. However, EAs typically require thousands of function evaluations to locate a near optimal solution. Hence when EAs are applied to problems involving high-fidelity simulation codes, the high computational costs involved pose a serious impediment to their successful application. This is primarily because a single exact fitness function evaluation (involving the analysis of a complex engineering system based on high-fidelity simulation codes) often consumes many minutes to hours or even days of CPU time. One promising way to significantly reduce the computational cost of EAs is to employ computationally cheap surrogate models in place of computationally expensive exact fitness evaluations [1], [2], [3], [4], [5]. By leveraging surrogate models, the computational burden can be greatly reduced since the efforts involved in building the surrogate model and optimization using it are much lower than the standard approach of directly coupling the simulation codes with the optimizer.

In this paper, we present a surrogate-assisted evolutionary optimization framework which combines both global and local surrogate models for solving computationally expensive problems. The present work is motivated by the lack of suitable multi-layer surrogate-assisted evolutionary optimization framework for solving computationally expensive problems. In other words, we show how multiple surrogate models can be combined to accelerate EA search. The first level of the proposed optimization framework involves a strategy that employs a Data Parallel Gaussian Process (DPGP) surrogate model to identify the promising individuals in the EA population. The DPGP approach was devised to reduce the high computational cost associated with Gaussian Process (GP) modeling [6]. Subsequently, the promising individuals undergo Lamarckian learning based on a trust-region enabled gradient-based search strategy that accelerates local search using computationally cheap Radial Basis Function (RBF) surrogate models. Lamarckian learning forces the genotype to reflect the result of improvement by replacing the locally improved individual back into the population to compete for reproductive opportunities.

The remainder of this paper is organized as follows: Section II presents a brief review of the surrogate assisted EAs described in the literature. Section III introduces the proposed surrogate-assisted evolutionary optimization framework for solving computationally expensive problems. Results obtained from numerical studies on a series of benchmark test functions are presented and discussed in Section IV. Section V presents the application of the proposed surrogate-assisted EA to a real-world aerodynamic shape design problem. Finally, section VI summarizes our main conclusions.

II. Related Work

Various techniques for the construction of surrogate models, often also referred to as metamodels or approximation models, have been used in engineering design optimization. Among these techniques, polynomial regression, Artificial Neural Network (ANN), Radial Basis Function (RBF), and Kriging (also referred to as Gaussian Process (GP) or Design and Analysis of Computer Experiments (DACE) models) are among some of the most prominent and commonly used techniques. Empirical studies of a number of these approximation methods have been made available recently. Among these methods, RBF and GP methods were shown to perform best under multiple modeling criteria in [7], [8], [9].

Apart from the techniques used to construct surrogate models, there has been a growing body of research focusing on the development of new EA frameworks for solving computationally expensive problems on a limited computational budget [2], [10], [11], [12]. Most existing approaches in this area replace the expensive exact fitness function with a global surrogate model of the fitness landscape constructed from a limited number of data points that hopefully mimics the entire search landscape. These data points are usually obtained during one or more generations of a classical evolutionary search. Subsequently, the surrogate model is updated online based on the new data points generated as the search evolves.

Keane and Petruzzelli [11] employed variable-fidelity analysis models in the context of genetic algorithm-based optimization of aircraft wings. Ratle [2] examined a simple strategy for integrating GAs with Kriging models. It uses a heuristic convergence criterion to determine when an approximate model should be updated. The same problem was revisited by El-Beltagy et al. [13], where the balance between the concerns of optimization with design of experiments was addressed. Jin et al. [14] coupled EAs with neural network-based surrogate and proposed an empirical criterion to switch between the expensive and approximate models during the search. In Song et al. [5], a real-coded GA coupled with Kriging was demonstrated on firtree structural optimization using a 3σ principle. A strategy for coupling EAs with local search based on a quadratic response surface model was considered in Liang et al. [15].

In practice, due to the *curse of dimensionality*, accurate global models become increasingly difficult to construct for problems with large numbers of variables. To circumvent these limitations, on-line local surrogate models have been considered in place of global models in the evolutionary search [1], [12]. Ong et. al. proposed a trust-region approach in the hybrid evolutionary search to interleave use of the exact objective and constraint functions with computationally cheap local surrogate models during Lamarckian learning [1]. Further the use of gradient information to improve the approximation accuracy of surrogate-assisted EAs was also considered in [12]. The local learning technique represents an instance of the transductive inference paradigm, which has recently been the focus of recent research in statistical learning theory [16], [17].

III. Evolutionary Optimization Framework Combining both Global and Local Surrogate Models

In this section, we present the essential ingredients of the proposed evolutionary optimization framework combining both global and local surrogate models for solving computationally expensive problems on a limited computational budget. In particular, we consider the general bound constrained nonlinear programming problem of the form:

$$\begin{aligned} \textbf{Minimize}: &\quad f(\mathbf{x}) \\ \textbf{Subject to}: &\quad \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \end{aligned} \tag{1}$$

where $f(\mathbf{x})$ is a scalar-valued objective function, $\mathbf{x} \in \mathbb{R}^d$ is the vector of continuous design variables, and $\mathbf{x}_l$ and $\mathbf{x}_u$ are vectors of lower and upper bounds, respectively.

In this work, we are interested in cases where the evaluation of $f(\mathbf{x})$ is computationally expensive, and it is desired to obtain a near optimal solution on a limited computational budget. It is worth noting that the present algorithm may be easily extended to constrained problems by adopting either an augmented Lagrangian or a penalty function approach. The readers are referred to the authors' earlier work in [1] on how this extension may be achieved.

For the sake of readability, we begin by outlining the proposed hierarchical surrogate-assisted evolutionary optimization framework in four phases:

Phase 0 $\{$*Initialization*$\}$: At the first step, a population of design points is initialized either randomly or using design of experiments techniques such as Latin hypercube sampling. These design points are evaluated using the exact objective function. The exact fitness values obtained are then archived in a central database together with the design vectors. After some initial period of time (for instance, after three generations of standard EA search), a Data Parallel Gaussian Process (DPGP) modeling method is devised to construct a surrogate model that represents the global trends of the entire fitness landscape, using the top ranking q archived design points of the database as training data.

Phase 1 $\{$*Global Search Strategy*$\}$: The first step in this phase is to check whether the DPGP global surrogate model needs be updated or not. If changes in the top ranking q design points of the database have taken place, the DPGP model will be updated using the new top q design points. In this manner, the computational cost can be reduced since the DPGP global surrogate model need not be reconstructed at every generation. Subsequently, the DPGP global surrogate model is used to pre-evaluate all individuals of the population. The predictions produced by using the DPGP model are used to pre-screen subsequent EA populations such that only the top ranking $\eta\%$ $(0 < \eta < 100)$ individuals undergo Lamarckian learning. This eliminates any unnecessary local searches from being conducted on individuals whose actual fitness is anticipated to be poor.

Phase 2 $\{$*Local Search Strategy*$\}$: A Lamarckian evolution process involving a trust-region framework devised for interleaving exact objective functions with computationally cheap RBF surrogate models is used during the gradient-based search. For each non-duplicated $\eta\%$ individuals, a local RBF surrogate model is built dynamically using only the m nearest neighboring data points in the central database. Each surrogate model represents the local fitness landscape in the vicinity of an individual and is hence termed here a local surrogate model. If an improved solution is found in the Lamarckian learning process, the genotype is forced to reflect the result of improvement by placing the locally improved individual back into the population to compete for reproductive opportunities. Subsequently, results of any new exact fitness obtained during the Lamarckian learning process are added into the central database, facilitating possible updating of surrogate models through online learning.

Phase 3 $\{$*Standard EA Operations*$\}$: The population then proceeds with the standard EA operators of crossover, mutation, etc. This process of hierarchical surrogate-assisted EA search is continued until the computational budget is exhausted or a user specified termination criterion

```
BEGIN
Initialize: Generate a database containing a population of designs.
Construct DPGP global surrogate model using top ranking $q$ design points in the database.
While (computational budget not exhausted)
    If (The top $q_{max}$ design points are changed.)
    * Update the DPGP global surrogate model using the top $q_{max}$ design points in the
      central database.
    End If
    Evaluate all individuals in the population using the DPGP global surrogate model.
    For each non-duplicated top ranking $\eta$ percent individual in the population
    * Apply trust-region enabled gradient-based local search strategy to the individual
      which interleaves the exact fitness function with a RBF local surrogate model for
      the fitness function.
    * Update the database with any new design points generated during the trust-region
      iterations together with the corresponding exact function values.
    * Replace the individuals in the EA population with the locally improved solution in
      the spirit of Lamarckian learning.
    End For
    Apply standard EA operators to create a new population.
End While
END
```

Fig. 1. Outline of the proposed evolutionary optimization framework combining both global and local surrogate models.

is met.

The basic steps of the proposed evolutionary optimization framework combining both global and local surrogate models for solving computationally expensive problems are listed in Figure 1. We next describe *Phases 1* and *2* in greater detail.

A. Global Search Strategy

The global search strategy is designed to identify search regions that contain better-quality solutions, here represented by the superior individuals in a EA population. An obvious and commonly used technique is to use a surrogate model to pre-evaluate the entire population of individuals based on the approximated fitness value [3], [18], [19] [20].

The choice of global surrogate model in the present framework should be one that is capable of modeling any complex global trends of the exact fitness landscape accurately. A statistically rigorous approximation is the idea of Bayesian interpolation or regression, which is also referred to as Gaussian process (GP) regression in the neural networks literature and Kriging in the geostatistics literature. It is generally recognized as a powerful tool for accurately modeling complex landscapes. Since GP regression possesses the aforementioned features, it make good sense to use it as a global surrogate model. Besides the mean fitness prediction, statistical error estimates can be readily obtained from the Gaussian Process approximation, which can be potentially exploited during evolutionary search; see, for example, [20]. In the present work, the Probability of Improvement (PoI) [21] predicted by the GP global surrogate model is used as the pre-selection criterion to pre-screen the population of promising individuals in our global search strategy. This may help to prevent premature convergence to a false global optimum, especially on multimodal and high dimensional problems. Nevertheless, a major disadvantage of the GP approximation method is that model construction, and in particular, hyperparameter tuning can be rather time-consuming when compared to other commonly used approximation methods.

We now briefly describe the GP regression technique used here for global surrogate model construction. In addition, the pre-selection criterion based on the probability of improvement is discussed.

Let $\mathcal{D} = \{\mathbf{x}_i, t_i\}, i = 1 \ldots n$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is an input design vector and $t_i \in \mathbb{R}$ is the corresponding target value. The GP surrogate model assumes the presence of an unknown true modeling function $f(\mathbf{x})$ and an additive noise term v to account for anomalies in the observed data. Thus:

$$t(\mathbf{x}) = f(\mathbf{x}) + v \tag{2}$$

The standard analysis requires the specification of prior probabilities on the modeling function and the noise model. From a stochastic process viewpoint, the collection $\mathbf{t} = \{t_1, t_2, ..., t_n\}$ is called a Gaussian process if every subset of $\mathbf{t}$ has a joint Gaussian distribution. More specifically,

$$P(\mathbf{t}|\mathbf{C}, \{x_n\}) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{t}-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{t}-\boldsymbol{\mu})\right) \tag{3}$$

where $\mathbf{C}$ is a covariance matrix parameterized in terms of hyperparameters $\boldsymbol{\theta}$, i.e., $\mathbf{C}_{ij} = k(x_i, x_j; \theta)$ and $\boldsymbol{\mu}$ is the process mean. The Gaussian process is characterized by this covariance structure since it incorporates prior beliefs both about the true underlying function as well as the noise model. In the present study, we use the following exponential covariance model

$$k(x_i, x_j) = e^{-(x_i-x_j)^T\Theta(x_i-x_j)} + \theta_{d+1} \tag{4}$$

where $\Theta = \mathrm{diag}\{\theta_1, \theta_2, ..., \theta_d\} \in \mathbb{R}^{d\times d}$ is a diagonal matrix of undetermined hyperparameters, and $\theta_{d+1} \in \mathbb{R}$ is an additional hyperparameter arising from the assumption that noise in the dataset is Gaussian (and output dependent). We shall henceforth use the symbol $\boldsymbol{\theta}$ to denote the vector of undetermined hyperparameters, i.e., $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_{d+1}\}$.

In practice, the undetermined hyperparameters are tuned to the data using the evidence maximization framework. Once the hyperparameters have been estimated from the data, predictions can be readily made for a new testing point. To illustrate this, assume that $\mathbf{t}_n$ represents the set of n targets, $\mathbf{C}_n$ the corresponding covariance matrix and that the process to be modeled has zero mean, i.e., $\boldsymbol{\mu} = 0$. Given a new point $\mathbf{x}_{n+1}$, it can be shown that the prediction t_{n+1} has a conditional probability distribution given by :

$$P(t_{n+1}|\mathcal{D}, \mathbf{C}, \mathbf{x}_{n+1}) = \frac{1}{Z}\exp\left(-\frac{(t_{n+1}-\hat{t}_{n+1})^2}{2\hat{\sigma}^2}\right) \tag{5}$$

where,

$$\hat{t}_{n+1} = \mathbf{k}_{n+1}^T(\mathbf{x})\mathbf{C}_n^{-1}\mathbf{t}_n \tag{6}$$

and

$$\sigma_{n+1}^2 = \mathbf{k}(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}; \boldsymbol{\theta})\mathbf{k}_{n+1}^T(\mathbf{x})\mathbf{C}_n^{-1}\mathbf{k}_{n+1} \tag{7}$$

where, $\hat{t}_{n+1}$ and σ_{n+1}^2 are the predicted posterior mean and variance, respectively, and $\mathbf{k}_{n+1} = \{k(x_{n+1}, x_1), k(x_{n+1}, x_2), \ldots, k(x_{n+1}, x_n)\} \in \mathbb{R}^n$. Hence, $\hat{t}_{n+1}$ is the mean prediction at point $\mathbf{x}_{n+1}$, σ_{n+1} is the standard deviation of t_{n+1} and provides a measure of the confidence at point $\mathbf{x}_{n+1}$. In other words, the Gaussian process approach results in a surrogate model which is a Gaussian random field.

From a computational perspective, the search for an optimal GP regressor under the evidence maximization framework [22] involves solving the following nonlinear maximum likelihood estimation (MLE) problem to determine the most probable hyperparameters $\boldsymbol{\theta}_{MP}$ for the given data:

$$\boldsymbol{\theta}_{MP} = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \quad \text{where} \quad L(\boldsymbol{\theta}) = -\frac{1}{2}\log\det\mathbf{C}_n - \frac{1}{2}\mathbf{t}_n^T\mathbf{C}_n^{-1}\mathbf{t}_N - \frac{n}{2}\log 2\pi \tag{8}$$

is the negative log likelihood function.

The main computational cost involved in constructing GP surrogate models occurs in the MLE phase. Since computing $L(\boldsymbol{\theta})$ and its gradient generally involves computing and decomposing a dense $n \times n$ covariance matrix ($\mathcal{O}(n^3)$ operations) at each iteration, training the GP model can be prohibitively expensive even for moderately sized data (e.g., say a few thousand data points). It is worth noting that an approximation method requiring high computational cost has limited utility in a surrogate-assisted optimization framework.

The computational bottleneck in standard GP modeling can be alleviated by employing a data parallel approach, which makes it possible to deal with datasets containing tens of thousands of points at modest computational cost [6]. Since a Gaussian stochastic process is completely specified by its covariance function, training a GP involves considering a parameterized covariance function and determining its hyperparameters $\boldsymbol{\theta}$ such that the log likelihood of the data is maximized. We next outline a compactly supported covariance function to facilitate data parallel GP learning.

To illustrate our approach, let us assume the existence of p disjoint and spatially localized subsets of the training data say $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p$. This partitioning of data can be readily achieved using the greedy load balancing clustering algorithm proposed by Choudhury et al. [6]. Given such a partitioning, the following covariance model can be employed to model the data.

$$\tilde{k}(\mathbf{x}_i, \mathbf{x}_j; c(\mathbf{x}_i), c(\mathbf{x}_j), \boldsymbol{\theta}) = \delta_{c(\mathbf{x}_i),c(\mathbf{x}_j)} k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}) \tag{9}$$

where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ are input vectors, δ_{ij} is the Kronecker delta function, $\boldsymbol{\theta}$ is a set of hyperparameters and $c : c(\mathbf{x}) \mapsto 1, 2, \ldots, p$ is an assignment function which maps the input point $\mathbf{x}$ to one of m available clusters. Then the covariance function in (8) can be immediately written for cluster i as :

$$\begin{aligned} \tilde{k}(\mathbf{x}_1, \mathbf{x}_2; c(.), \boldsymbol{\theta}) = & \quad k(\mathbf{x}_1, \mathbf{x}_2; \boldsymbol{\theta}), \quad c(\mathbf{x}_i) = c(\mathbf{x}_j) = i & (10) \\ = & \qquad 0, \qquad\quad otherwise & (11) \end{aligned}$$

where $\boldsymbol{\theta}_i$ denotes the set of hyperparameters for the local model trained on the ith cluster. Consider the case when $p = 2$, i.e, when the data has been partitioned into two disjoint spatially localized subsets. Then using (10), the covariance matrix can be written as:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{22} \end{bmatrix} \tag{12}$$

where $\mathbf{K}_{ii} \in \mathbb{R}^{n_i \times n_i}$ contains correlation terms explicitly from the ith cluster which consists of n_i points. Since in this case the determinant of the covariance matrix K can be written as the product of determinants of the blocks $\mathbf{K}_{11}$ and $\mathbf{K}_{22}$, the log likelihood can be split into individual log likelihoods for the two partitions, i.e.,

$$L(\boldsymbol{\theta}) = L(\boldsymbol{\theta}_1) + L(\boldsymbol{\theta}_2). \tag{13}$$

From the preceding discussion, it is clear that the use of a compactly supported covariance function naturally leads to a data parallel learning approach to GP regression and hence provides a means to handle large datasets. In general, it is often the case that the predictive capability may reduce when an increasing number of clusters are used [6]. However, this degradation in performance is often very small and acceptable given the significant savings in computational cost.

As mentioned earlier, the Gaussian process approach results in a random field approximation of the analysis code. Using the output mean prediction $\hat{t}(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x})$ of GP model, a variety of pre-selection criteria for the selection of promising individuals may be formulated to accelerate evolutionary optimization search. An obvious and common pre-selection criterion is to use the mean prediction for exploiting the knowledge of the GP model to find the promising individuals. However, this may lead to premature convergence in many cases due to the inevitable limitations on the accuracy of a global surrogate model constructed using a few data points. Hence, there is also a need to explore new areas of search space for a more thorough global search. To circumvent this problem, Torczon and Trosset [23] proposed minimizing the merit function $f_M = \hat{t}(\mathbf{x}) - \alpha\sigma(\mathbf{x})$. The first term in the merit function ensures exploration of regions in the design space that are likely to have better solutions whereas the second term favors those points at which the predictions are likely to have maximum error. The parameter α in some sense balances the tradeoff between local exploitation and global exploration. An alleged disadvantage of this approach is that the user has to choose an appropriate fixed value of α or develop a sensible strategy for adapting this parameter as the search progresses.

Here we consider using the Probability of Improvement (PoI) instead of the merit function since previous work in [19] suggests that it performs well. To illustrate the approach used here, consider the case when it is aimed to solve a minimization problem. Let t^- denote the smallest value of all the outputs in the training dataset used to construct the GP surrogate. Subsequently, it is intended to use the surrogate model to predict a new point $\mathbf{x}^*$ at which the output is likely to be lower than t^-. The PoI at the point $\mathbf{x}^*$ (i.e., the probability that the surrogate prediction at $\mathbf{x}^*$ is lower than t^-) can be readily computed from the posterior mean $\hat{t}(\mathbf{x}^*)$ and standard deviation $\sigma(\mathbf{x}^*)$ as follows:

$$PoI(\mathbf{x}^*) = \Phi\left(\frac{t^- - \hat{t}(\mathbf{x}^*)}{\sigma(\mathbf{x}^*)}\right) \tag{14}$$

where $\Phi(.)$ is the normal cumulative distribution function.

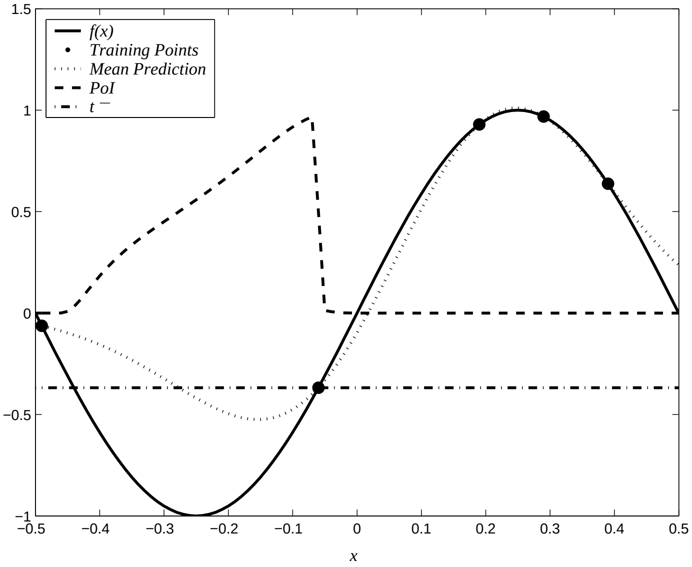


Fig. 2. Characteristics of the PoI criterion when the GP model is trained on points generated by an one-dimensional function.

Figure 2 shows the characteristics of the PoI pre-selection criterion for a one-dimensional test function. It may be noted from the figure that the PoI criterion is able to correctly identify the region in which the true objective function must be sampled to drive $f(x)$ below t^-. The points identified by maximizing $PoI(\mathbf{x})$ can be appended to the baseline training dataset to update the surrogate model (and consequently the probability of improvement criterion). Increasing the number of training points in such a stage-wise fashion improves the ability of the PoI criterion to correctly locate the region in which the optimum lies. Note here that this statistical criterion is only used to filter the individuals in an EA population – as discussed later, a local search strategy is employed to identify the best solution in the vicinity of an individual. We also mention here the possibility of employing alternative statistical measures such as the expected improvement criterion proposed by Jones et al. [21].

B. Local Search Strategy

The local search strategy is designed to work with a locally trained system that adjusts to the local properties of the training data in each area of the input space. The surrogate model is constructed using only the m neighboring data points in the database nearest to the design point of interest because the neighboring points are likely to have more impact than remote ones [3].

The surrogate model used by the local search strategy is built dynamically for every filtered and non-duplicated individual. Since local surrogate models will probably be built thousands of times during the overall search, computational efficiency is a major concern. This consideration motivates the use of RBF local surrogate models, which can be efficiently applied to approximate multiple-input multiple-output data, particularly when a few hundred data points are used for training. The RBF model also has found to offer reasonable accuracy as well as fast training. Since computational efficiency is the major concern, the RBF model is suitable for the local search strategy of the proposed optimization framework.

Let $\mathcal{D} = \{\mathbf{x}_i, t_i\}, i = 1 \ldots n$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ and $t_i \in \mathbb{R}$ are the input and output, respectively. Then the local surrogate models are interpolating radial basis function networks of the form

$$t(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(||\mathbf{x} - \mathbf{x}_i||), \tag{15}$$

where $K(||\mathbf{x} - \mathbf{x}_i||) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a RBF and $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \in \mathbb{R}^n$ denotes the vector of weights.

Typical choices for the kernel include linear splines, cubic splines, multiquadrics, thin-plate splines, and Gaussian functions [24]. We propose the use of linear splines, i.e. $||x - c_i||$, to construct surrogate models since our earlier study [1] suggests that this kernel is capable of providing models with good generalization capability at a low computational cost. Further, our local search strategy embeds a Feasible Sequential Quadratic Programming optimizer (FSQP) within a trust-region framework, which ensures convergence to the local optimum of the exact computationally expensive fitness function [1], [25]. More specifically, for each non-duplicated individuals among the top ranking $\eta\%$ in the population, the local search strategy proceeds with a sequence of trust-region subproblems of the form

$$\textbf{Minimize}: \quad \hat{f}^k(\mathbf{x} + \mathbf{x}_c^k) \tag{16}$$

$$\textbf{Subject to}: \quad ||\mathbf{x}|| \leq \Omega^k \tag{17}$$

where $k = 0, 1, 2, \ldots, k_{max}$, $\hat{f}(x)$ is the approximation function corresponding to the objective function $f(x)$. $\mathbf{x}_c^k$ and Ω^k are the starting point and the trust-region radius used for local search at iteration k, respectively.

For each subproblem (or during each trust-region iteration), surrogate models of the exact fitness function, viz., $\hat{f}^k(\mathbf{x})$ are created dynamically. The m nearest neighbors of the initial point, $\mathbf{x}_c^k$, are extracted from the archived database of design points evaluated so far using the exact analysis codes. The criterion used to determine the similarity between design points is the simple Euclidean distance metric. These points are then used to construct local surrogate models of the exact fitness function.

The surrogate models thus created are used to facilitate the necessary fitness function estimations in the local searches. During local search, we initialize the trust-region Ω using the minimum and maximum values of the design points used to construct the surrogate models. After each iteration, the trust-region radius Ω^k is updated based on a measure which indicates the accuracy of the surrogate model at the kth local optimum, $\mathbf{x}_{lo}^k$. After computing the exact values of the fitness function at this point, the figure of merit, ρ^k, is calculated as

$$\rho^k = \frac{f(\mathbf{x}_c^k) - f(\mathbf{x}_{lo}^k)}{\hat{f}(\mathbf{x}_c^k) - \hat{f}(\mathbf{x}_{lo}^k)}. \tag{18}$$

The above equations provide a measure of the actual versus predicted change in the exact fitness function values at the kth local optimum. The value of ρ^k is then used to update the trust-region radius as follows [26]:

$$\begin{aligned}\Omega^{k+1} &= 0.25\Omega^k, \quad \text{if } \rho^k \leq 0.25, \\ &= \Omega^k, \quad \text{if } 0.25 < \rho^k \leq 0.75, \\ &= \xi\Omega^k, \quad \text{if } \rho^k \geq 0.75,\end{aligned} \tag{19}$$

where $\xi = 2$, if $||\mathbf{x}_{lo}^k - \mathbf{x}_c^k||_\infty = \Omega^k$ or $\xi = 1$, if $||\mathbf{x}_{lo}^k - \mathbf{x}_c^k||_\infty < \Omega^k$.

The trust-region radius, Ω^k, is reduced if the accuracy of the surrogate, measured by ρ^k is low. Ω^k is doubled if the surrogate is found to be accurate and the kth local optimum, $\mathbf{x}_{lo}^k$, lies on the trust-region bounds. Otherwise the trust-region radius remains unchanged.

The exact solutions of the fitness functions at the kth local optimum are combined with the existing neighboring data points to generate new surrogate models in the subsequent trust-region iterations. The initial point for iteration $k+1$ is defined by

$$\begin{aligned}\mathbf{x}_c^{k+1} &= \mathbf{x}_{lo}^k, \quad \text{if } \rho^k > 0 \\ &= \mathbf{x}_c^k, \quad \text{if } \rho^k \leq 0.\end{aligned} \tag{20}$$

The trust-region process for an individual terminates when the maximum number of trust-region iterations permissible, k_{max}, chosen by the user, is reached. Lamarckian learning then proceeds if the k_{max} local optimum solution obtained is an improvement over that of the initial individual.

IV. Performance Analysis

In this section, we analyze the performance of the proposed evolutionary optimization framework. Since a Genetic Algorithm (GA) is used here in the empirical studies, we also refer to the algorithm proposed in the present work as Surrogate-Assisted Genetic Algorithm with Global and Local search Strategy (SAGA-GLS). We evaluate the performances of the SAGA-GLS algorithm against a traditional GA, and two surrogate-assisted evolutionary optimization algorithms that were recently introduced in the literature [1], [3]. These are representatives of Surrogate-Assisted Evolutionary Algorithm with Global-search Strategy (SAGA-GS) or Local-search Strategy (SAGA-LS).

At each search generation, the SAGA-GS employs the standard RBF or GP surrogate model to screen the entire population of individuals. The predefined top ranking $\eta\%$ individuals in the EA population then undergo exact evaluations. In contrast to [3], the SAGA-GS we employed in our study involves using the computationally cheap DPGP and estimates the ranking of the individuals based on their probability of improvements rather than merely using the mean prediction. On the other hand, the SAGA-LS we considered corresponds to the earlier work of the authors [1] that evolves the solution of each individual in the spirit of Lamarckian learning using local RBF surrogates.

A standard GA is employed with population size of 50, uniform crossover and mutation operators at probabilities 0.6 and 0.01, respectively. A stochastic universal sampling algorithm is used for selection. However, apart from the standard GA settings, the two user-specified parameters of the SAGA-LS are – (1) maximum number of nearest neighboring data points used to construct the local surrogate model m_{max} and (2) maximum trust region iterations k_{max}. In our numerical studies, we set m_{max} and k_{max} to 100 and 3, respectively. In SAGA-GS and SAGA-GLS, the maximum number of training design points (i.e., q_{max}) and clusters for constructing the global surrogate model using DPGP are configured as 2000 and 4, respectively. It is worth noting that in the surrogate assisted algorithms, all design points in the database will be used for constructions of global or local surrogate models, if the training design points are lower than the maximum number configured, i.e., q_{max} or m_{max}. In addition, all configurations used in this study were values suggested in earlier studies [1], [6], [27].

The results obtained from our empirical studies on a range of benchmark test functions, i.e., two unimodal test functions (Sphere and Rosenbrock test function) and three multimodal test functions (Ackley, Griewank and Rastrigin test function) are presented in Figures 3-7. All benchmark test functions used in the study are of 20 dimensions and have a single global minimum at zero (see appendix I for greater details of the test functions). Note that the results presented are averaged over 20 simulation runs conducted with a limited computational budget of (6×10^3) exact objective function evaluations.

From the results obtained in Figures 3-7, it is clear that all the surrogate-assisted evolutionary optimization algorithms considered here are capable of searching more efficiently than the standard GA on the benchmark problems under a limited computational budget. Further, both SAGA-LS and SAGA-GLS appear to converge much faster and yield improved solution quality as compared to SAGA-GS on all the benchmark problems. This makes sense since Memetic algorithms, i.e., EAs that employ local search heavily such as SAGA-LS and SAGA-GLS, are generally well-known to search more effectively and efficiently. The superiority of SAGA-LS and SAGA-GLS are more evident on the unimodal benchmark problems.

It is worth nothing that SAGA-GLS converges significantly faster than the SAGA-LS on the unimodal problems. For instance, we observed that SAGA-GLS converges correctly to the global minima of the exact objective function in Figure 3 within the limited computational budget. This outcome may be easily explained. Since the sphere problem is a smooth, symmetric function and unimodal, it makes perfect sense to use the Lamarckian learning process in SAGA-GLS or SAGA-LS involving any gradient-based local search. However, in contrast to SAGA-LS, only the $\eta\%$ top ranking individuals among the entire EA population in SAGA-GLS undergo the Lamarckian learning process, thus providing significant computational cost savings.

Consider next the complex multimodal benchmark problems. On multimodal functions, the number of local minima increases exponentially with the problem dimensions, often they present hills and valleys with misleading local optima. Any gradient-based optimization algorithm would easily get stuck in a local minima. Hence, performance studies of surrogate-assisted EAs on multimodal problems reflect the algorithm's ability to escape from poor local optima and head towards the global optimum. Figures 5-7 illustrate the search performances of GA, SAGA-GS, SAGA-LS and SAGA-GLS on the Ackley, Griewank and Rastrigin multimodal benchmark test functions, respectively. From these figures, SAGA-GLS is once again demonstrated to accelerate the evolutionary search significantly faster than GA, SAGA-GS or SAGA-LS on all of the multimodal problems considered. For the Ackley function, we observed that the SAGA-GLS is capable of converging correctly to the global minima of the exact objective function even though there are thousands of local minima in the entire search space (see Figure 5). This indicates the robustness of the SAGA-GLS to premature convergence.

Overall, the results obtained also imply that the SAGA-GLS is not only capable of identifying the better quality individuals in each EA population (via its global search strategy), at the same time, its intrinsic local search strategy can also exploit these filtered individuals effectively and efficiently. This combination of the global and local search strategies in the SAGA-GLS is the key reason for the improvements in search quality at a significantly lower computational budget than existing surrogate-assisted EAs.

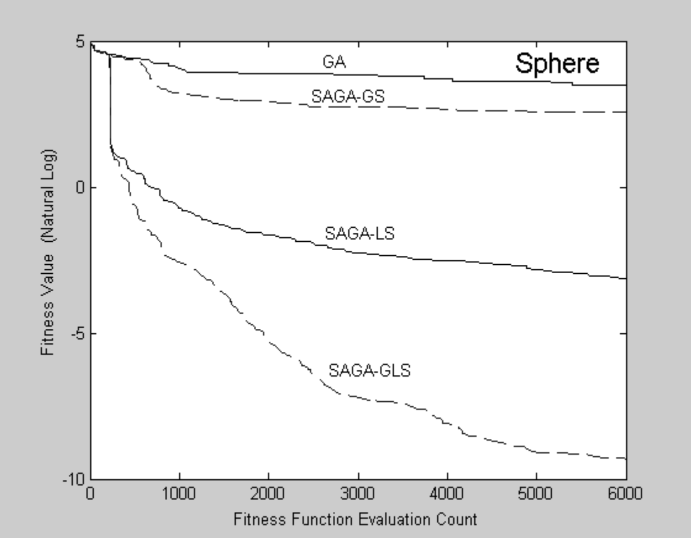


Fig. 3. Convergence trends of the GA, SAGA-GS, SAGA-LS and SAGA-GLS framework for Sphere function

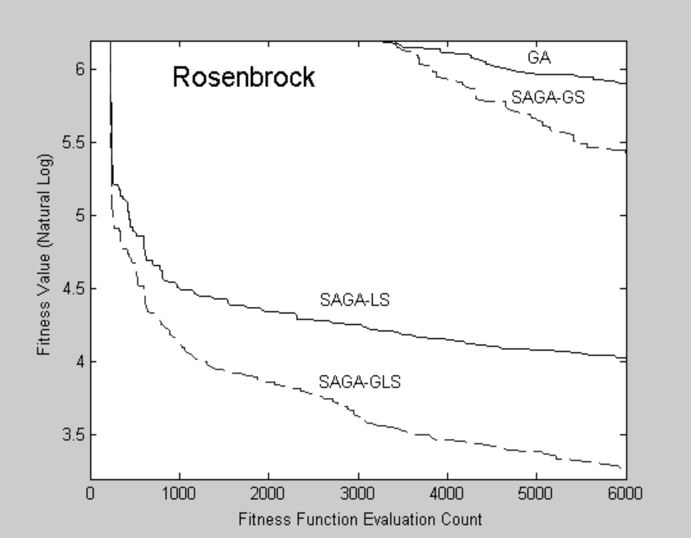


Fig. 4. Convergence trends of the GA, SAGA-GS, SAGA-LS and SAGA-GLS framework for Rosenbrock function

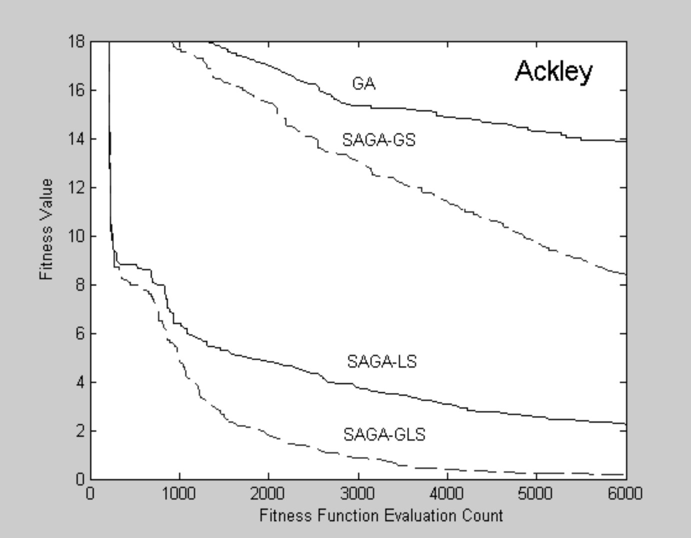


Fig. 5. Convergence trends of the GA, SAGA-GS, SAGA-LS and SAGA-GLS framework for Ackley function

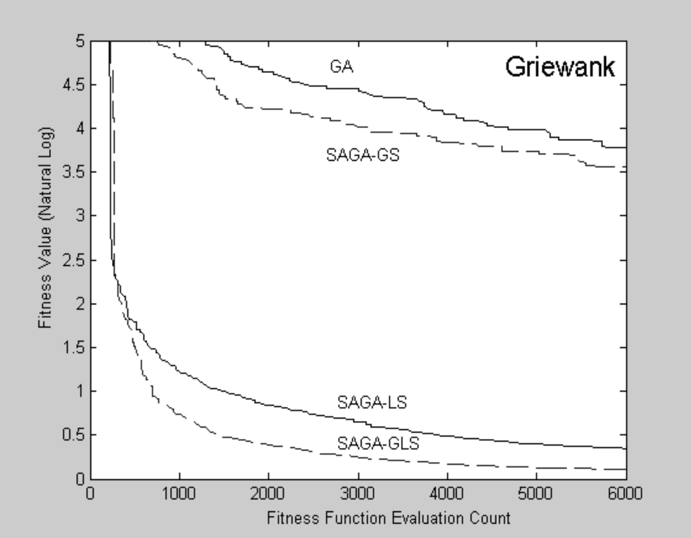


Fig. 6. Convergence trends of the GA, SAGA-GS, SAGA-LS and SAGA-GLS framework for Griewank function

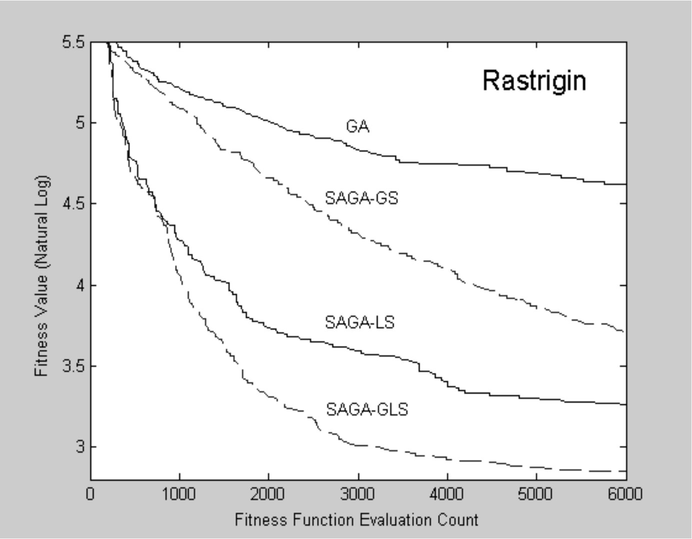


Fig. 7. Convergence trends of the GA, SAGA-GS, SAGA-LS and SAGA-GLS framework for Rastrigin function

V. Aerodynamic Shape Design Optimization

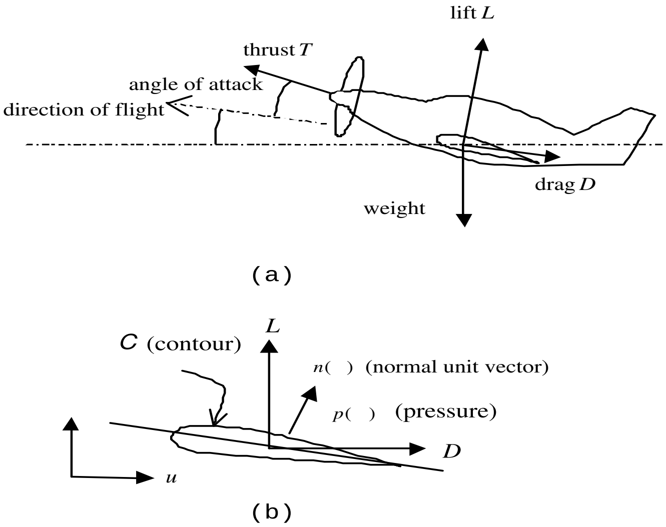
In this section, we apply the SAGA-GLS to efficient aerodynamic shape design. In particular, we consider the parametric design optimization of a 2D airfoil structure with minimum drag-over-lift ratio, D/L.

The drag D and lift L on an airplane are the components of the total aerodynamic force parallel and vertical to the direction of flight, respectively, as shown in Figure 8a. The importance of the ratio D/L in design can be understood, for example, in two airplane performance considerations [28]. First, the engine thrust required for level and unaccelerated flight, i.e. cruise, is given by

$$T_{cruise} = (weight\ of\ aircraft) \times D/L \tag{21}$$

Second, an airplane in a power-off gliding flight will descent at an angle - $\theta_{gliding}$ gliding given by

$$\tan\theta_{gliding} = D/L \tag{22}$$



(a)

(b)

Fig. 8. Forces acting on: (a) an airplane, and (b) an airfoil

In both cases, it is obvious that the smaller the ratio D/L, the better the performance. In the first case, a small ratio means less engine power is required for cruising flight, thus saving fuel. In the second case, low drag over lift entails a safer gliding flight in the case of engine failure. While the drag and lift forces on an airplane are determined by various body components, the contribution of the wings is dominant. This motivates the development of an approach for designing airfoil geometries by minimizing the D/L ratio.

In an airfoil shape optimization problem using computation fluid dynamics, the drag and lift forces can be obtained by calculating the flow field around the airfoil under prescribed operating conditions, defined by the Mach number which represents the incident flow rate, and the angle of attack (see Figure 8b). Ignoring friction, the flow is governed by the 2D Euler equations:

$$\frac{\partial w}{\partial t} + \frac{\partial f_1}{\partial z_1} + \frac{\partial f_2}{\partial z_2} = 0 \tag{23}$$

with t as the time variable,

$$w = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho E \end{bmatrix}, f_1 = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ \rho u_1 H \end{bmatrix}, f_2 = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 \\ \rho u_2 H \end{bmatrix} \tag{24}$$

where ρ is the density, u_1 and u_2 are the flow velocity components in the Cartesian space with coordinates z_1 and z_2, p is the pressure, E is the total specific energy and H is the total specific enthalpy. Moreover, the pressure is given by $p = (\gamma - 1)\rho(E - \frac{1}{2}u_1^2 - \frac{1}{2}u_2^2)$, where γ is the specific heat [29].

Thus, the drag D and lift L are simply the components opposite the direction of flight $\vec{u}_\infty$, and the direction perpendicular to flight $\vec{\tau}_\infty$, respectively, of the resultant force due to pressure acting along the contour $\mathcal{C}$ of the airfoil (see Figure 8b). They are given by the following integrals:

$$D = \oint_{\mathcal{C}} p(\sigma)\vec{n}(\sigma).\vec{u}_\infty d\sigma \tag{25}$$

and

$$L = \oint_{\mathcal{C}} p(\sigma)\vec{n}(\sigma).\vec{\tau}_\infty d\sigma. \tag{26}$$

A. Optimization Setup

The optimization problem considered here is to achieve an airfoil design for an optimized drag-to-lift ratio profile for constant operating conditions of Mach 0.5 and Angle of Attack, $AOA = 2.0$ degrees. The geometry of the airfoil is represented using 24-parameter Hicks-Henne functions as illustrated in Figure 9 [30].

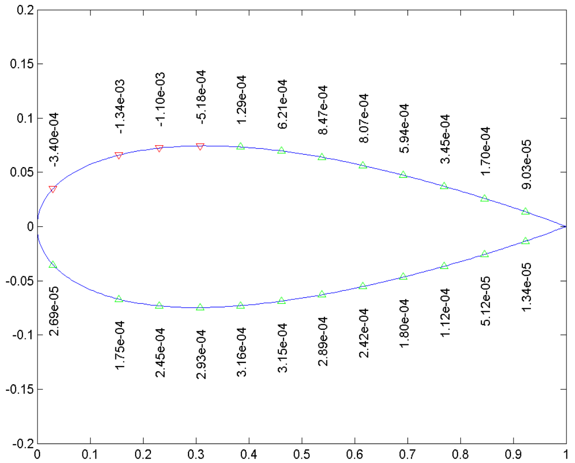


Fig. 9. Airfoil geometry characterized using 24-parameter Hicks-Henne functions.

For the airfoil problem we consider, a single exact adjoint CFD analysis takes approximately 30 minutes to compute. In comparison, surrogate model construction using linear splines RBF takes merely a fraction of a second to compute while building the DPGP model takes less than a minute on a typical workstation. When dealing with computationally expensive problems that cost many minutes of cpu time per function evaluation, this training cost may be regarded as insignificant.

We conduct the parametric design of the airfoil using all three evolutionary optimization frameworks, i.e., standard GA, SAGA-LS and SAGA-GLS. It is worth noting that SAGA-GL algorithm was omitted for the sake of brevity since it has been shown as inferior to SAGA-LS and SAGA-GLS. Apart from using a population size of 20 (due to the immense computational cost), all other parameters are kept the same as in Section IV.

B. Optimization Results

The design histories of the aerodynamic 2D airfoil optimization problem using standard GA, SAGA-LS and SAGA-GLS frameworks are presented in Figure 10. Using a population of 20 initial design points based on Latin hypercube sampling, these designs are evaluated using the exact Adjoint CFD analysis code. All three EA frameworks proceed with the standard GA operations using the exact CFD analysis code for the first three generations. Hence, they share the same search history at the initial search phase. This initial phase represents the period where the SAGA-LS and SAGA-GLS forms its database of past design points for constructing surrogate models later during search.

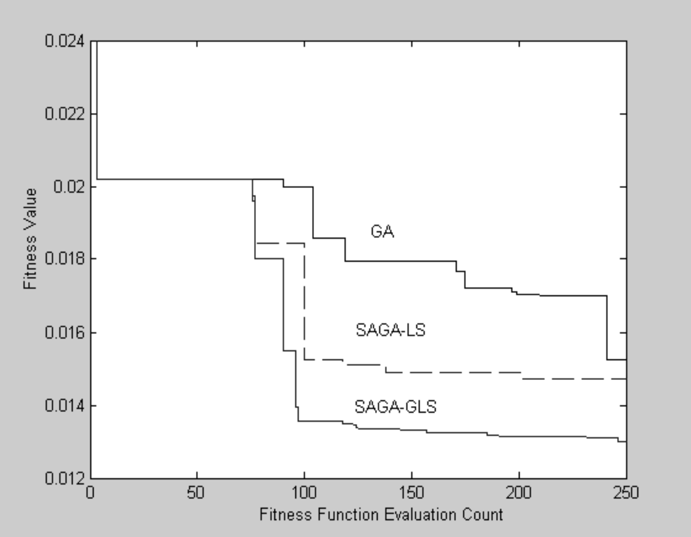


Fig. 10. Convergence trends of the GA, SAGA-LS and SAGA-GLS framework for Aerodynamic Shape Design problem.

Clearly, the results in Figure 10 indicate that both SAGA-GLS and SAGA-LS arrived at better airfoil designs than the standard GA, using significantly lower computational costs. Moreover, SAGA-GLS was shown to accelerate the evolutionary search much faster as compared to both standard GA and SAGA-LS, producing improved design much earlier.

VI. Conclusion

For computationally expensive optimization problems, the use of surrogate models helps to greatly reduce the number of evaluations of the exact fitness model by exploiting the information contained in the search history. In this paper, we present a novel surrogate-assisted evolutionary optimization framework that combines both global and local surrogate models and exploits the integration between them. The algorithms make use of the global surrogate model and a probability of improvement pre-selection criterion to rank the promising individuals in the EA population. A surrogate-assisted Lamarckian learning approach is then applied to these promising individuals to accelerate evolutionary search.

Experimental studies are presented for a number of unimodal and multimodal benchmark test functions to study the effect of changing various user-specified parameters introduced in this framework. Results are also presented for a real-world aerodynamic shape design problem. The empirical results were compared with those obtained using a standard GA and other surrogate-assisted EAs. The results obtained suggest that the proposed optimization framework is capable of solving computationally expensive optimization problems more efficiently than the standard GA, SAGA-GS and SAGA-LS on a limited computational budget.

Appendix I
Test Problems

A. Sphere Test Function

$$f(x) = \sum_{i=1}^{n}(x_i^2) \tag{27}$$

$$-5.12 \leq x_i \leq 5.12, i = 1, 2, \ldots, n.$$

B. Rosenbrock Test Function

$$f(x) = \sum_{i=1}^{n-1}(100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \tag{28}$$

$$-2.048 \leq x_i \leq 2.048, i = 1, 2, \ldots, n-1.$$

C. Ackley Test Function

$$f(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum\limits_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum\limits_{i=1}^{n}\cos 2\pi x_i} \tag{29}$$

$$-32.768 \leq x_i \leq 32.768, i = 1, 2, \ldots, n.$$

D. Griewank Test Function

$$f(x) = 1 + \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n}\cos(x_i/\sqrt{i}) \tag{30}$$

$$-600 \leq x_i \leq 600, i = 1, 2, \ldots, n.$$

E. Rastrigin Test Function

$$f(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i)) \tag{31}$$
$$-5.12 \leq x_i \leq 5.12, i = 1, 2, \ldots, n.$$

ACKNOWLEDGMENT

This research work is a collaboration between Nanyang Technological University, University of Southampton and Temasek Laboratories. It is supported by an NTU/SCE grant number CE-SUG 3/03. In addition, the authors would like to thank the following research centers, Parallel and Distributed Computing Center, Center for Multimedia and Network Technology in Nanyang Technological University and Computational Engineering and Design Center in the University of Southampton for their support in this work.

REFERENCES

[1] Y. S. Ong, P. B. Nair, and A. J. Keane, “Evolutionary optimization of computationally expensive problems via surrogate modeling,” *American Institute of Aeronautics and Astronautics Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[2] A. Ratle, “Accelerating evolutionary algorithms using fitness function models,” *Parallel Problem Solving from Nature*, vol. 5, pp. 87–96, 1998.

[3] K. C. Giannakoglou, “Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence,” *International Review Journal Progress in Aerospace Sciences*, vol. 38, no. 5, pp. 43–76, 2002.

[4] M. K. Karakasis, A. P. Giotis, and K. C. Giannakoglou, “Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization,” *International Journal for Numerical Methods in Fluids*, vol. 43, pp. 1149–1166, 2003.

[5] W. B. Song, “Shape optimisation of turbine blade firtrees,” Ph.D. dissertation, Univ. of Southampton, 2002.

[6] A. Choudhury, P. B. Nair, and A. J. Keane, “A data parallel approach for large-scale gaussian process modeling,” in *Proc. the Second SIAM International Conference on Data Mining*, Arlington, VA, Apr. 2002.

[7] A. A. Guinta and L. Watson, “A comparison of approximation modelling techniques: polynomial versus interpolating models,” in *Proc. the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-98-4758*, St. Louis, MO, USA, Sept. 1998.

[8] D. D. Daberkow and D. N. Marris, “New approaches to conceptual and preliminary aircraft design: A comparative assessment of a neural network formulation and a response surface methodology,” in *AIAA, 1998 World Aviation Conference*, Florida , USA, Sept. 1998.

[9] R. Jin, W. Chen, and T. W. Simpson, “Comparative studies of metamodeling techniques under multiple modeling criteria,” *Structural and Multidisciplinary Optimization*, vol. 23, no. 1, pp. 1–13, 2001.

[10] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.

[11] A. J. Keane and N. Petruzzelli, “Aircraft wing design using ga-based multi-level strategies,” in *Proc. the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-2000-4937*, Long Beach, CA, USA, Sept. 2000, pp. A00–40 171.

[12] Y. S. Ong, K. Y. Lum, P. B. Nair, *et al.*, “Global convergence unconstrained and bound constrained surrogate-assisted evolutionary search in aerodynamic shape design,” in *Proc. IEEE Congress on Evolutionary Computation CEC’03, Special Session on Design Optimisation with Evolutionary Computation*, vol. 3, Canberra , Australia, Dec. 2003, pp. 1856–1863.

[13] M. A. El-Beltagy, P. B. Nair, and A. J. Keane, “Metamodelling techniques for evolutionary optimization of computationally expensive problems: Promise and limitations,” in *Proc. IEEE the Genetic and Evolutionary Computation Conference (GECCO’99)*, Florida , USA, July 1999, pp. 196–203.

[14] Y. Jin, M. Olhofer, and B. Sendho, “A framework for evolutionary optimization with approximate fitness functions,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 5, pp. 481–494, 2002.

[15] K. H. Liang, X. Yao, and C. Newton, “Evolutionary search of approximated n-dimensional landscapes,” *International Journal of Knowledge-Based Intelligent Engineering Systems*, vol. 4, no. 3, pp. 172–183, 2000.

[16] O. Chapelle, V. Vapnik, and J. Weston, “Transductive inference for estimating values of functions,” *Advances in Neural Information Processing Systems*, vol. 12, 1999.

[17] V. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, 1998.

[18] D. B$\ddot{u}$che, N. N. Schraudolph, and P. Koumoutsakos, “Accelerating evolutionary algorithms with gaussian process fitness function models,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Special Issue on Knowledge Extraction and Incorporation in Evolutionary Computation*, 2004.

[19] H. Ulmer, F. Streichert, and A. Zell, “Evolution startegies assisted by gaussian processes with improved pre-selection criterion,” in *Proc. IEEE Congress on Evolutionary Computation CEC’03*, Canberra, Australia, Dec. 2003, pp. 692–699.

[20] M. Emmerich, A. Giotis, M. Oezdemir, K. Giannakoglou, and T. Baeck, “Metamodel-assisted evolution strategies,” *PPSN VII, LNCS 2439*, pp. 361–370, 2002.

[21] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.

[22] D. Mackay, “Bayesian interpolation,” *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992. [Online]. Available: citeseer.ist.psu.edu/article/mackay91bayesian.html

[23] V. Torczon and M. W. Trosset, “Using approximations to accelerate engineering design optimization,” in *Proc. of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 98-4800*, no. TR-98-33, St. Louis, Missouri, USA, Sept. 1998. [Online]. Available: citeseer.ist.psu.edu/torczon98using.html

[24] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[25] J. F. Rodriguez, J. E. Renaud, and L. T. Watson, “Convergence of trust region augmented lagrangian methods using variable fidelity approximation data,” *Structural Optimization*, vol. 15, no. 3-4, pp. 141–156, 1998.

[26] N. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon, “A trust region framework for managing the use of approximation models in optimization,” *Structural Optimization*, vol. 15, no. 1, pp. 16–23, 1998.

[27] Z. Zhou, Y. S. Ong, and P. B. Nair, “Hierarchical surrogate-assisted evolutionary optimization framework,” in *Proc. IEEE Congress on Evolutionary Computation CEC’04, Special Session on Learning and Approximation in Design Optimization*, Portland, USA, June 2004, pp. 1586–1593.

[28] A. D. J. Anderson, *Introduction to Flight, 4^{th} edition*. McGraw-Hill, 2000.

[29] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics – The Finite Volume Method*. Longman Group Ltd, 1995.

[30] R. M. Hicks and P. A. Henne, “Wing design by numerical optimization,” *Journal of Aircraft*, vol. 15, no. 7, pp. 407–412, 1978.