We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

4

A Memetic Algorithm Assisted by an Adaptive Topology RBF Network and Variable Local Models for Expensive Optimization Problems

Yoel Tenne and S.W. Armfield
School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Australia

1. Introduction

A common practice in modern engineering is that of **simulation-driven optimization**. This implies replacing costly and lengthy laboratory experiments with **computer experiments**, i.e. computationally-intensive simulations which model real world physics with high fidelity. Due to the complexity of such simulations a single simulation run can require up to several hours of CPU time of a high-performance computer [45, 56, 61].

With computer experiments the simulation-driven optimization process is cast as a nonlinear optimization problem having three distinct features:

- There is typically no analytic expression for the relation between inputs (candidate designs) and outputs, i.e. it is a **black-box function**.
- Each simulation run is expensive so only a small number (~ 200) of runs can be made.
- The underlying real-world physics and/or numerical solution often yield an inputs–output landscape which is multimodal and nonsmooth.

A promising approach to tackle such problems is the **surrogate-assisted memetic optimization**. A memetic algorithm combines an evolutionary algorithm (EA) with an efficient local search so as to obtain both efficient exploration and exploitation during the optimization search [21, 65]. A surrogate-model is a computationally cheaper mathematical approximation of the expensive objective function and is used during the optimization search in lieu of the expensive function [2, 45] (in some references the term **metamodel** is used synonymously while 'surrogate-model' is reserved for a lower-fidelity simulation [42, 87]). Thus, using surrogate-models circumvents the problem of simulation cost and allows evaluation of many candidate designs.

In this study we propose a surrogate-assisted memetic algorithm which builds upon recent advances in computational intelligence and optimization [9, 53, 60, 83–85, 94]. The proposed algorithm aims to address four open issues:

- Obtaining a global model with a small generalization error is too expensive: analysis has shown the number of sites required to achieve a fixed generalization error grows exponentially with the problem dimension [79]. To avoid allocating all function evaluations to the global model we employ **a combination of global and local surrogate-models** to achieve an efficient optimization search.

Source: Advances in Evolutionary Algorithms, Book edited by: Witold Kosiński, ISBN 978-953-7619-11-4, pp. 468, November 2008, I-Tech Education and Publishing, Vienna, Austria

- The accuracy of a global Lagrangian model can degrade due to over-fitting: a Lagrangian model learns the exact features of the data which can lead to over-fitting and degrades its generalization ability. To address this we use as a global surrogate-model an **artificial neural network based on a RBF network (RBFN) with an adaptive network topology**. We describe an efficient method for adapting and training the network.
- Convergence to a false optimum: the local search relies on local models, hence if these are badly inaccurate the local search may converge to a false optimum. To address this we employ a **trust-region framework applied to general nonlinear local models**. Such models can describe a complicated landscape better than the quadratic models of the classical trust-region approach. We propose a framework for safeguarding and improving the models' accuracy.
- Difficulty in selecting an optimal model: different models can be used during the local search, e.g. RBF and Kriging. Due to lack of information the user typically chooses an inoptimal model which degrades the local search performance. To address this we describe **a method for model selection based on an approximate generalization error**. The method results in local models which vary during the local search.

Accordingly, in this chapter we propose a framework of memetic optimization using variable global and local surrogate-models for expensive optimization problems. To obtain a global model with good generalization ability it uses an RBFN artificial neural network. During the local search it makes an extensive use of accuracy assessment to select the local models and to improve them if necessary. It also employs the trust-region approach but replaces the quadratic models with the more general RBF and Kriging models. Rigorous performance analysis shows the proposed algorithm outperforms several variants of a reference surrogate-assisted EA.

This chapter is organized as follows: Sect. 2 reviews related work and Sect. 3 describes in detail the proposed algorithm. This is followed by Sect. 4 which provides the performance analysis and lastly Sect. 5 summarizes this chapter.

2. Related work

2.1 Expensive optimization problems

Since EAs require many function evaluations to converge several approaches have been studied so as to make them applicable to expensive optimization problems.

One such approach is **fitness inheritance**, where only a fraction of the offspring are evaluated with the computationally expensive objective function and the rest inherit their fitness from their parents [32, 75].

A second approach is that of **hierarchical** or **variable-fidelity** optimization which uses several computer simulations of varying computational cost (fidelity); promising candidate solutions migrate from low- to high-fidelity simulations and vice versa [15, 68, 71].

A third approach, which we adapt in this study, is that of **surrogate-assisted** optimization [2, 20, 26, 30, 53, 63, 77, 83, 85, 94]. As mentioned, a surrogate-model is a mathematically-cheaper approximation of the expensive function (typically an interpolant). A least-squares quadratic model (originally designed for real-world experiments which are noisy) are used in the Response Surface Methodology [5, 48]. Recent studies have used neural-networks [29, 61], Kriging [63, 72] and radial basis functions [85, 94]. The framework of surrogate-assisted

optimization also involves the design of computer experiments [25, 73] and accuracy assessment of surrogate-models [42, 74].

2.2 Memetic optimization

Heuristics using random processes, such as EAs, are efficient in exploring the objective function landscape and can escape non-global optima. However, in late stages the optimization search focuses on a small subset of the search space so exploiting the local function behavior is preferred. This motivates the hybridization of random-based heuristics with efficient local search algorithms to balance **exploration–exploitation**, i.e. an efficient global and local search [88]. Within the framework of evolutionary optimization such algorithms are termed **hybrid algorithms** or **memetic algorithms**.

Examples include hybridization of an EA with a quasi-Newton and conjugate directions algorithms [21, 62, 66] and various direct search methods [33, 65, 91, 92]. Multiobjective memetic algorithms were studied in [19, 61] and a parallel algorithm was studied in [10]. An algorithm for selection among candidate local searchs was studied in [52]. Memetic algorithms aimed for expensive optimization problems were studied in [53, 54, 83, 84, 93, 94].

3. The proposed algorithm

3.1 Initialization and main loop

Analysis shows the number of sites required to achieve a fixed interpolation error grows exponentially with problem dimension [79]. This implies it is inefficient to allocate most or even all function evaluations to a single model as this may still result in an inaccurate model. Accordingly, we use a sequential approach where we only aim for a coarse global model and then use the remaining function evaluations to converge to an optimum [87]. As such, the algorithm begins by generating a Latin Hypercube sample (LHS) of $N_0 = 0.2fe_{\max}$ where $fe_{\max}$ is the prescribed limit on evaluations of the expensive function. This provides a space-filling sample which improves the model accuracy [41, 73]. The sites are evaluated with the true objective function to obtain their corresponding responses and both are copied into a cache which is initially empty. Next, a global model is generated based on all cached sites using the procedure described in Sect. 3.2. We then search for an optimum of this model using a memetic algorithm. Lastly in the optimization iteration, a local search is initiated from the predicted optimum so as to converge to an optimum of the expensive function, as described in Sect. 3.4. The main loop terminates when the number of function evaluations reaches the prescribed limit $fe_{\max}$ ($fe_{\max}$ = 100, 150 and 200 were used for performance analysis). A pseudocode of the main algorithm is given in Algorithm 1.

Algorithm 1: A pseudocode of the main loop.

```
generate initial sites with LHS and evaluate them;
copy sites into the cache;
while fe ⩽ fe_max do
    generate a global model based on the cache;
    perform a local search;
    update cache;
```

3.2 A variable-topology RBFN global model

A global model which is a Lagrangian interpolant, i.e. satisfying the conditions of exact interpolation

$$\mathcal{S}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)\,,\ i = 1 \ldots n\ (= \text{sample size})\,, \tag{1}$$

can suffer from two demerits: a) it can generalize poorly due to over-fitting to the given data [4, 7, 34] and b) it can become computationally-expensive (since it accounts for all sites) and numerically unstable (due to ill-conditioning) [6, 11, 28].

To circumvent these issues we use for the global model an artificial neural network with radial basis functions neurons (processing units), a design termed an **RBF network** (RBFN). Such networks have two merits: a) both theoretical analysis and real-world experience have shown they generalize well [22, 43, 59, 81] and b) they have a simpler topology compared to other networks and hence are more easily implemented and trained [46, 57, 58].

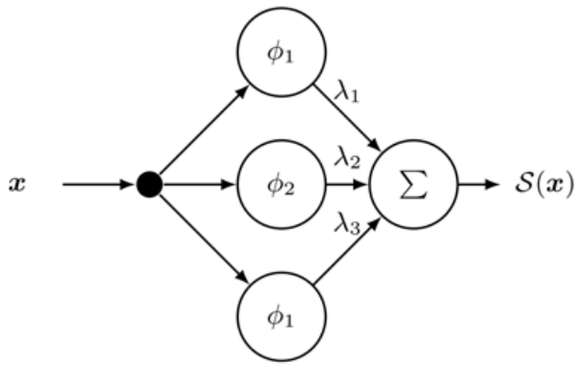


Fig. 1. An RBFN with three neurons (processing units).

Figure 1 shows a diagram of a typical RBFN. It comprises of three layers: the input layer, the processing layer comprised of neurons and the output layer which is a weighted sum of the neuron responses. An RBFN generalizes well and avoids over-fitting since it generates an abstraction of the data set. This is achieved by using fewer neurons than sample sites (so the centres of the neuron RBFs typically do not coincide with any of the data sites) and careful training of the network parameters. The response of an RBFN is given as

$$\mathcal{S}(\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{t}_j\|_2}{c_j}\right)\,, \tag{2}$$

where N is the number of neurons, λ_j is a coefficient, $\boldsymbol{t}_j$ is a basis-function (or kernel) centre and c_j is a shape parameter (or hyper-parameter). The neurons are RBF Gaussian functions which assist in modelling nonlinear functions [22, 43, 49, 57].

To avoid ill-conditioning and expensive calculation the network needs to be compact (minimizing the number of neurons N) while still be capable of generalizing well. Also, it is difficult to prescribe an optimal topology so the network should be self-adaptive [18, 30, 31, 39, 58]. Accordingly, we implement such a self-adaptive network which operates as follows. Initially, the data set is split into a training set ($\mathcal{X}_{\text{tra}}$) and a testing set ($\mathcal{X}_{\text{tra}}$) which are disjoint (we use a 80–20 training–testing ratio). Starting from a single neuron, the network is trained

with $\mathcal{X}_{\text{tra}}$ and is tested with $\mathcal{X}_{\text{tes}}$, an approach termed **holdout** [23, 82]. The generalization error is measured by the normalized root mean square error (NRMSE) over $\mathcal{X}_{\text{tes}}$, i.e.

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{|\mathcal{X}_{\text{tes}}|} (\mathcal{S}(\boldsymbol{x}_i) - f(\boldsymbol{x}_i))^2}{\text{Var}(f(\boldsymbol{x}_i))}}\,, \tag{3}$$

where $\boldsymbol{x}_i$ is the ith site in the testing set $\mathcal{X}_{\text{tes}}$ and the numerator is the sum of the Gaussian loss-function (or discrepancy)

$$L(\boldsymbol{x}) = (\mathcal{S}(\boldsymbol{x}) - f(\boldsymbol{x}))^2 \tag{4}$$

over then training set [34]. The denominator is the variance of the responses in the testing set. Besides the NRMSE the loss-function values over the training and testing set are also calculated, i.e.

$$L_{\text{tes}} = \sum_{i=1}^{|\mathcal{X}_{\text{tes}}|} (\mathcal{S}(\boldsymbol{x}_i) - f(\boldsymbol{x}_i))^2 \tag{5}$$

and similarly for the training set yielding L_{tra}. If NRMSE > NRMSE$^{\ulcorner}$ where NRMSE$^{\ulcorner}$ is prescribed than $0.1|\mathcal{X}_{\text{tra}}|$ neurons are added to the network and the new network is trained as explained below. The network stops growing if NRMSE ≤ NRMSE$^{\ulcorner}$ or if the number of neurons equals the number of training sites ($N = |\mathcal{X}_{\text{tra}}|$). After the network stopped growing the chosen topology is that which had the lowest weighted error

$$e_{\text{w}} = 0.8L_{\text{tes}} + 0.2L_{\text{tra}}\,, \tag{6}$$

where a larger weight is given to the testing error over the training error.
For each number of neurons the network parameters (RBF centres, coefficients, shape parameters) need to be trained to achieve good generalization. While it is possible to train the network in a fully supervised manner by minimization of the generalization error convergence is slow [46]. Accordingly, we implement a fully unsupervised learning where the RBF centres are obtained by a k-means clustering algorithm [31, 46], the shape parameters are obtained from

$$c_j = \max\{0.1\bar{d},\, 1\}\,, \quad j = 1 \ldots N\,, \tag{7}$$

where $\bar{d}$ is the mean l_2 distance between all sites in the data set $\mathcal{X}$ (related to the Gaussian rate of decay) [57, 58]), and the coefficients λ are obtained from the normal least-squares equations

$$\boldsymbol{\Phi}^{\text{T}}\boldsymbol{\Phi}\boldsymbol{\lambda} = \boldsymbol{\Phi}^{\text{T}}\boldsymbol{f}\,, \tag{8}$$

where $\boldsymbol{f}$ is the vector of responses and $\boldsymbol{\Phi}$ is the interpolation matrix

$$\boldsymbol{\Phi} : \Phi_{i,j} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{t}_j\|_2}{c_j}\right). \tag{9}$$

Figure 2 shows an example of a model training with the variations in L_{tra} and L_{tes}. When the network is over-trained the testing error begins to grow. The parameters are taken from the cycle which minimized e_{w} before over-training. Figure 3 shows an example of the adaptation of the proposed RBFN. Algorithm 2 gives a pseudocode of the proposed algorithm for the adaptive RBFN.

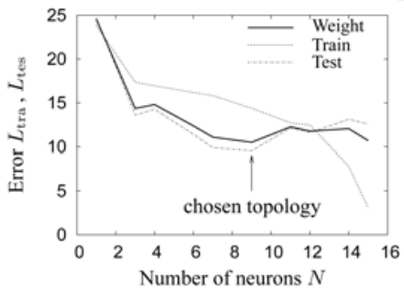
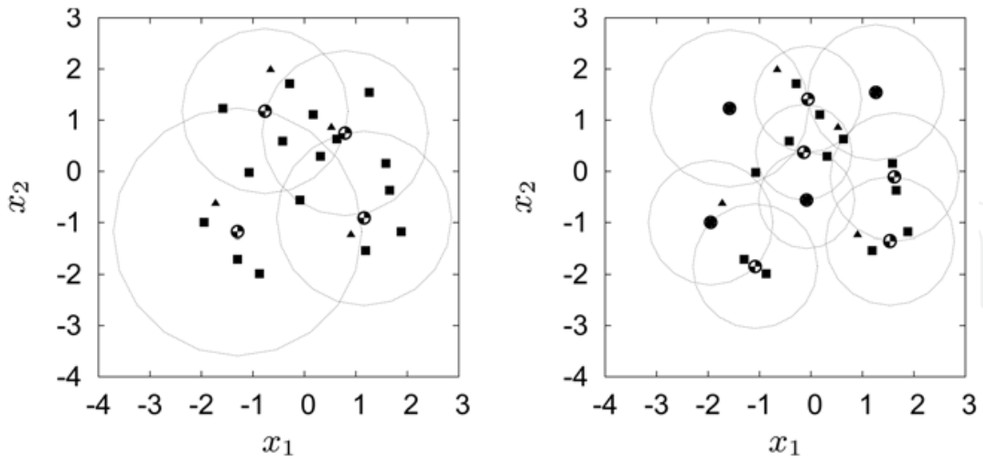


Fig. 2. An example of the RBFN training with the Rastrigin-5D objective function. As the number of neurons increases both training error and testing error decrease until overtraining commences at 9 neurons (indicated by an increase in the testing error). The chosen topology has the minimal weighted error.



(a) 4 neurons (b) 9 neurons

Fig. 3. An example of the RBFN topology adaptation with the Rastrigin-2D function. A sample of 20 sites was split into training (■) and testing (▲) sites. We show each topology by its RBFN centres (◓) and the corresponding shape parameters (the radius of the circles).

Algorithm 2: The proposed algorithm for the adaptive topology RBFN

inputs
  $\mathcal{X}$; /* set of sites and responses */

split $\mathcal{X}$ into a training set and a testing set which are disjoint;
set number of neurons $N = 1$;
repeat
  train network;
  calculate NRMSE and weighted error;
  if $NRMSE > NRMSE^{\star}$ **then**
    increase number of neurons;
until $NRMSE \leqslant NRMSE^{\star}$ *or (number of neurons is* $|\mathcal{X}_{\mathrm{tra}}|$ *)* ;
select network topology yielding the smallest weighted error;

3.3 Memetic search for an optimum of the global model

After generating the global model $\mathcal{S}(\boldsymbol{x})$ we use a memetic algorithm to search for an optimum of it. The memetic algorithm first employs a real-coded EA [8] for efficient exploration. The EA uses a population size $\boldsymbol{s}_{\mathrm{pop}} = 50$, linear ranking, stochastic universal sampling (SUS), intermediate recombination, elitism with a generation gap $g_{\mathrm{gap}} = 0.9$ and the breeder-genetic-algorithm mutation operator with probability $p_{\mathrm{m}} = 0.05$ [47]. The evolutionary search is stopped when no improvement is observed after $g_{\mathrm{n.i.}} = 10$ generations; the small setting for $g_{\mathrm{n.i.}}$ is since we do not require the EA to converge to a very accurate solution, as this is accomplished by the following step. The optimum found by the EA is then used as the initial solution for an SQP solver which uses the finite-differences quasi-Newton BFGS algorithm. This yields $\boldsymbol{x}_{\mathrm{c}}$ an improved predicted optimum of the global model. During the memetic optimization stage approximate function values are obtained from the surrogate-model (the objective function is not used).

3.4 The local search

Since the global model is coarse $\boldsymbol{x}_{\mathrm{c}}$ may be a bad approximation to a true optimum of the expensive function. Accordingly, we use $\boldsymbol{x}_{\mathrm{c}}$ as an initial guess for a local search to search for a true optimum. Two considerations with the local search are efficiency (which suggests using local models requiring fewer sites than the global model) and accuracy (which suggests using a procedure to safeguard against convergence to a false optimum). Both of these goals are accomplished by using a trust-region approach, as described below. To further improve the local search we propose a method for selecting the model type (as either RBF or Kriging) and to improve the models, if necessary; this results in local models which vary during the local search.

3.4.1 A trust-region approach

The classical trust-region approach generates at each iteration a quadratic model and obtains its constrained optimum (a truncated Newton step) as a quadratic programming problem. However, such models cannot adequately describe a complicated or multimodal landscape so instead we generate more flexible local models (either RBF or Kriging) and obtain their

constrained optimum (in the trust-region) using a memetic search. The trust-region framework safeguards the model accuracy and ensures convergence to an optimum of the expensive objective function, i.e. it is a **framework for managing models** [1, 12, 68].

The initial trust-region is taken as a cuboid centred at $\boldsymbol{x}_c$ the predicted optimum of the global model and is of size Δ (with an initial size $\Delta_0 = 0.1$), i.e.

$$\mathcal{T} = \{\boldsymbol{x} : \|\boldsymbol{x}_c - \boldsymbol{x}\|_\infty \leqslant \Delta\}\,. \tag{10}$$

All cached sites which are in the trust-region are used to generate the local surrogate-model. We exclude remote sites to emphasize only the local function behaviour.

The model type is selected using the algorithm described in Sect. 3.2 and the constrained optimum of the local model in $\mathcal{T}$, $\boldsymbol{x}_m$, is obtained by the memetic search described in Sect. 3.3.

Following the classical trust-region approach the predicted optimum is evaluated with the true objective function and a merit value is calculated

$$\rho = \frac{f(\boldsymbol{x}_m) - f(\boldsymbol{x}_c)}{\mathcal{S}(\boldsymbol{x}_m) - \mathcal{S}(\boldsymbol{x}_c)}\,, \tag{11}$$

where $\mathcal{S}(\boldsymbol{x}_m)$ now denotes the current local surrogate-model.

A main difference to the classical trust-region framework is that the latter assumes the quadratic model is accurate (i.e. based on an exact gradient and Hessian) while here we also need to account for model inaccuracy due to the interpolation on a finite set. As such, the model may be inaccurate due to an insufficient number of sites in the trust-region. Reducing the trust-region size too quickly due to model inaccuracy can lead to premature termination of the local search [9]. To avoid this we relate the model accuracy to the number of sites in the trust-region, denoted as s. A reasonable criterion to consider the model accurate is when $s \geq d + 1$ (d being the problem dimension). This threshold is based on the number of sites required to model the gradient of the objective function (and hence to identify a descent direction) by well-established methods like quasi-Newton finite-differences or polynomial interpolation [9]. However, if the allowed number of function evaluations fe_{max} is small and the problem dimension is high too many sites are needed to consider the model accurate. Accordingly, we use the threshold value $s^{\lceil} = \min\{d + 1\,,\, 0.1 fe_{max}\}$.

Based on ρ, s and $s^{\lceil}$ the proposed algorithm performs one of the following updates:

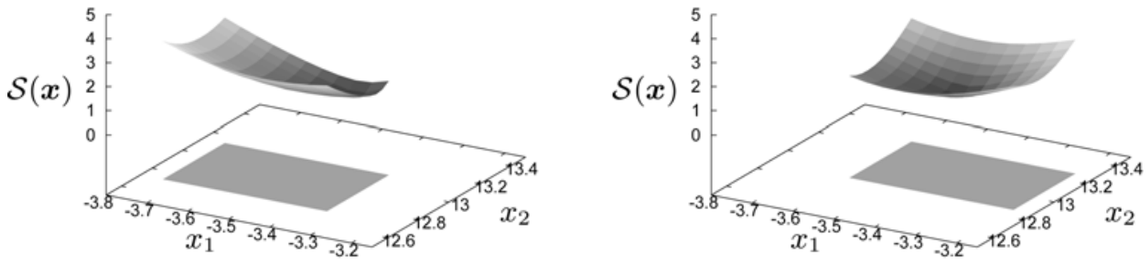
- if $\rho > 0$: then the surrogate-model is accurate since a better solution has been found. Following the classical trust-region framework we centre the trust-region at the new optimum $\boldsymbol{x}_m$ and increase the trust-region size by a factor δ_+.
- if $\rho \leq 0$ and $s < s^{\lceil}$: the local model is inaccurate but this is attributed to an insufficient number of sites in the trust-region. Thus we improve the accuracy of the local model in the trust-region by adding a site using the model improvement algorithm (Sect. 3.4.3).
- if $\rho \leq 0$ and $s \geq s^{\lceil}$: the local model is based on a sufficient number of sites but fails to predict an improvement due to the trust-region size. Following the classical trust-region framework we decrease the trust-region size by a factor δ_-.

After the model and trust-region have been updated the current local search iteration is finished. The local search is stopped if the trust-region is small enough $\Delta < \Delta_{\min}$ (we use $\Delta_{\min} = \Delta_0 \cdot \delta^2$) or if the number of evaluations of the true objective function exceeds $fe_{\max}$.

Some additional comments on the local search:

- At most only two evaluations of the true function are performed at each local search iteration.
- All sites evaluated during the local search are added to the cache for later use.

Figure 4 shows an example of a local search with the proposed trust-region approach used with the Branin function. A pseudocode of the proposed trust-region local search is given in Algorithm 3.



(a) Iteration 2 (b) Iteration 5

Fig. 4. An example of the trust-region local search using local models (RBF or Kriging). The objective function is Branin. For iterations 2 and 5 the chosen model (Kriging) and the corresponding trust-region are shown.

Algorithm 3: A pseudocode of the local search.

inputs

$\mathcal{X}$; /* set of sites */
$\mathcal{Y}$; /* set of responses */
$\boldsymbol{x}_c$; /* predicted optimum of global model */

$s^\star = \min\{d+1\,,\, 0.1fe_{\max}\}$; /* model accuracy threshold */

repeat

define a cuboid trust-region with a face size Δ and centred at $\boldsymbol{x}_c$;
find the cached sites inside the trust-region;
generate a local surrogate-model using these sites (use model selection);
use a memetic search to find $\boldsymbol{x}_m$ the optimum of the local model in the trust-region;
calculate a trust-region merit value ρ;
perform a trust-region update:

$$\begin{cases} \rho \geqslant 0 & \boldsymbol{x}_c \leftarrow \boldsymbol{x}_m\,, \text{increase } \Delta \\ \rho < 0 \bigcap s < s^\star & \text{improve local model (add a site)} \\ \rho < 0 \bigcap s \geqslant s^\star & \text{decrease } \Delta \end{cases}$$

until $\Delta < \Delta_{\min}$ *or* $fe \geqslant fe_{\max}$;

3.4.2 Model selection

To assist the optimization search we wish to generate a surrogate-model which is optimal, i.e. as accurate as possible. We select among two candidate models, namely radial basis functions (RBFs) or Kriging, as these have performed well in benchmark tests against other models [17, 25, 73, 74].

The RBF surrogate-model is a Lagrangian interpolant which is a linear combination of basis functions. To ensure the non-singularity of the interpolation matrix we consider an RBF model which uses linear basis functions [44] such that

$$\mathcal{S}(\boldsymbol{x}) = \sum_{i=1}^{n} \lambda_i \phi_i(\boldsymbol{x}) \,, \quad \phi_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_i\|_2 \,, \tag{12}$$

where $\boldsymbol{n}$ is the number of sites, $\phi_i(\boldsymbol{x})$ are the linear radial basis functions and the coefficients λ_i are obtained from the linear system

$$\boldsymbol{\Phi\lambda} = \boldsymbol{f} \,. \tag{13}$$

A Kriging (or a spatial-correlation) model uses a global 'drift' function o which a stationary Gaussian process is overlaid; the former captures the global trend while the latter provides local adjustments [40, 45, 69]. We adapt the common approach where the drift function is taken as constant (e.g. set to 1) so the model is given by

$$\mathcal{S}(\boldsymbol{x}) = \beta + Z(\boldsymbol{x}) \,, \tag{14}$$

where β is the drift function coefficient and $Z(\boldsymbol{x})$ is the Gaussian process function [45, 69]. The Gaussian process is assumed to have a mean zero and variance σ. Deviating from the random error approach of the Response Surface Methodology, the response at any site is considered correlated with other sites. The correlation between two sites $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is defined by a covariance function

$$C(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma^2 R(\boldsymbol{x}_1, \boldsymbol{x}_2) \,, \tag{15}$$

where $R(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is a prescribed spatial correlation function (SCF). Following [45] we consider the exponential SCF

$$R(\boldsymbol{x}_1 \,, \boldsymbol{x}_2) = \prod_{j=1}^{d} \exp\left(-\theta |\boldsymbol{x}_{1\,,j} - \boldsymbol{x}_{2\,,j}|\right) \,. \tag{16}$$

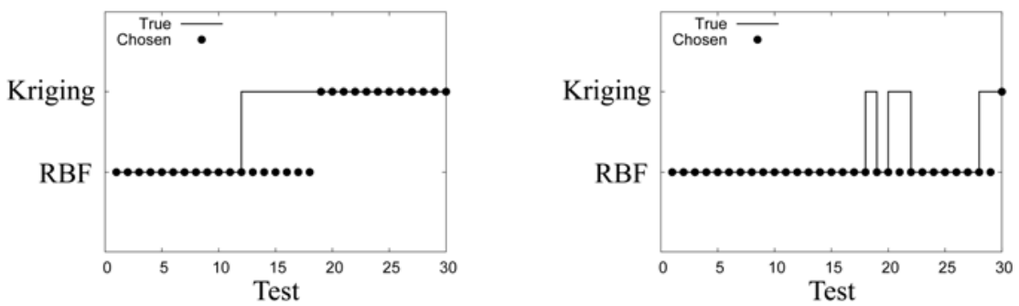
The Kriging model is defined once β and θ have been fixed. For a given data set the value of θ is obtained by maximum likelihood estimation [37]. Having found the optimal θ and assuming a constant drift function then the Kriging model is

$$\mathcal{S}(\boldsymbol{x}) = \hat{\beta} + \boldsymbol{r}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{R}^{-1} (\boldsymbol{f} - \boldsymbol{1}\hat{\beta}) \,, \tag{17}$$

where R is the correlation matrix for data set $\mathcal{X}$, $\boldsymbol{r}$ is the correlation vector between $\boldsymbol{x}$ and $\mathcal{X}$ and $\hat{\beta}$ is the least-squares estimate of β

$$\hat{\beta} = (\mathbf{1}^{\mathrm{T}}\boldsymbol{R}^{-1}\mathbf{1})^{-1}\mathbf{1}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{f}\,. \tag{18}$$

Details of the Kriging code implementation are given in [78].
The two different possible models, namely linear RBF and Kriging, introduces the issue of **model selection**. To assist the local search we wish to select the most accurate model, i.e. having the least generalization error. Similarly to Section 3.2 we approximate the generalization error based on the available data set. While it is possible to use the holdout method for approximating the generalization error a better estimate is obtained if repeated models are generated and all sites are used both for training and for testing, an approach known as the **leave-one-out cross-validation** (LOOCV) [42, 80]. The estimate is obtained as follows: given a candidate model (in our case a linear RBF or Kriging) then for each site $\boldsymbol{x}_i$, $i = 1\ldots n$ a surrogate-model is generated using all sites except $\boldsymbol{x}_i$ and the Gaussian loss-function of this surrogate-model is calculated at $\boldsymbol{x}_i$. The estimated generalization error is then the mean of all observed errors. The model corresponding to the smallest LOOCV error is assumed to be the most accurate. In this basic form the LOOCV procedure requires generating n surrogate-models, which is expensive. To circumvent this, for the RBF we use an efficient procedure proposed in [67] while for the Kriging we use a procedure proposed in [45].



(a) Rosenbrock-10D (b) Rastrigin-20D

Fig. 5. Examples of the model selection algorithms. The solid line (—) indicates which model was more accurate based on a large sample of 250 sites while the dot (•) indicates which model was selected by the proposed method based on a small sample.

Figure 5 shows two examples of the proposed model selection algorithm. The following procedure was repeated 30 times to obtain statistically significant results. We used the Rosenbrock-10D and Rastrigin-20D test functions and 50 sites generated by LHS. The proposed method was used to select between an RBF model and a Kriging model. A separate testing sample of 250 LHS sites was used to obtain a more accurate estimate of the true generalization error of the models. It follows the proposed method selects (in the large majority of cases) the model whose true generalization error is indeed smaller.
The outcome of the model selection is that the proposed memetic algorithm uses **variable surrogate-models** (either linear RBF or Kriging) during the local search. A pseudocode of the model selection algorithm is given in Algorithm 4.

Algorithm 4: A pseudocode for model selection.

inputs
  $\mathcal{X}$; /* set of sites */
  $\mathcal{Y}$; /* set of responses */

for *model type = linear RBF, Kriging* **do**
  calculate LOOCV error for the model;
select the model having the smaller LOOCV error.

3.4.3 Model improvement

If the local model is deemed inaccurate, i.e. there is an insufficient number of sites in the trust-region then a new site is generated to improve the model accuracy (reduce its generalization error). Analysis of various surrogate-models (polynomial, RBF and Kriging) relates their generalization error to the distribution of sites [9, 28, 70]. Clustered sites do not provide sufficient information on the objective function and lead to an ill-conditioned interpolation matrix which further degrades the model accuracy. The distribution of sites is measured by the **fill distance**

$$h = \sup \min \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \,, \tag{19}$$

i.e. the radius of the largest ball in the feasible domain $\mathcal{F}$ which does not contain any sites in its interior [36, 70]. To improve the model accuracy (increase h) new sites should be added such that they are remote from existing sites. Thus, to improve the model in the trust-region we seek a site which maximizes the fill distance for the augmented set $\{\mathcal{X}_\mathcal{T} \bigcup \boldsymbol{x}_i\}$ where $\mathcal{X}_\mathcal{T}$ is the set of sites in the trust-region. To obtain the model-improving site $\boldsymbol{x}_i$ we formulate the nonlinear optimization problem

$$\begin{aligned} &\max \min\{\|\boldsymbol{x} - \boldsymbol{x}_j\|_2\} \,, \quad j = 1 \ldots |\mathcal{X}_\mathcal{T}| \\ &\text{s.t. } \boldsymbol{x} \in \mathcal{T} \end{aligned} \tag{20}$$

We solve (20) by generating an initial sample of candidate sites and starting an SQP solver from the best one (having the maximum separation distance). This results in sites distributed similarly to the **maximin design** [27]. After $\boldsymbol{x}_i$ has been found it is evaluated with the true objective function and is added to the cache. A pseudocode of the model improvement iteration is given in Algorithm 5.

Algorithm 5: A pseudocode for model improvement

inputs
  $\mathcal{X}$; /* set of sites */

search for a site $\boldsymbol{x}_\mathrm{i} \in \mathcal{T}$ which maximizes the fill distance of $\{\mathcal{X}_\mathcal{T} \bigcup \boldsymbol{x}\}$;
evaluate $\boldsymbol{x}_\mathrm{i}$ and add $\boldsymbol{x}_\mathrm{i}$, $f(\boldsymbol{x}_\mathrm{i})$ to the cache;

3.5 Additional remarks

In this section we provide additional remarks on the complete algorithm.

- The local search is initiated only if the distance of the predicted optimum $\boldsymbol{x}_c$ from all cached sites is larger than $\Delta_{\min}$
- As the cache grows the interpolation matrix $\boldsymbol{\Phi}$ becomes ill-conditioned and this degrades the solution accuracy of (8) [28]. To circumvent this we solve (13) by the numerically stable **truncated singular value decomposition (TSVD)** such that

$$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{V} = \boldsymbol{\Phi}\,, \tag{21}$$

where $\boldsymbol{\Sigma}$ is the diagonal matrix of singular values σ_i of $\boldsymbol{\Phi}$. Given the responses vector $\boldsymbol{f}$ and defining

$$\boldsymbol{C} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{f} \tag{22}$$

and a vector $\boldsymbol{y}$ such that

$$\boldsymbol{y} : y_i = \begin{cases} c_i/\sigma_i & \sigma_i \geqslant \epsilon_{\mathrm{SVD}} \\ 0 & \sigma_i < \epsilon_{\mathrm{SVD}} \end{cases} \tag{23}$$

the solution vector is

$$\boldsymbol{\lambda} = \boldsymbol{V}\boldsymbol{y}\,. \tag{24}$$

Thus the solution vector is generated by the span of the vectors corresponding to a sufficiently large singular value. We use $\epsilon_{\mathrm{SVD}} = 10\epsilon$, where ϵ is the machine precision.

4. Performance analysis

We assessed the performance of the proposed algorithm using both mathematical test functions and a real-world problem of airfoil shape optimization. In these tests the proposed algorithm was also benchmarked against two variants of a reference surrogate-assisted EA which is representative of many others [64]; Algorithm 6 gives its pseudocode.

Algorithm 6: A pseudocode of the reference surrogate-assisted EA.

```
generate initial sites with LHS and evaluate them;
copy sites into the cache;
generate a global model based on the cache;
while fe ⩽ fe_max do
    search for model optimum using an EA for 10 generations;
    evaluate top 10% of elites with true objective function;
    add elites to cache;
    update global model and population fitness;
```

The two variants differ by the surrogate-model they use, namely either a linear RBF model or a Kriging model with an exponential spatial correlation function.

	General parameters	
$fe_{\max}$	max. (true) objective function evaluations	200
N_0	sample size for initial global model	$0.2 \cdot fe_{\max}$
	RBFN global model	
δ_{h}	holdout ratio	0.2
NRMSE*	threshold NRMSE	0.1
	Memetic algorithm	
s_{pop}	population size	50
g_{gap}	generation gap	0.9
$g_{\max}$	maximum generations	20
p_{m}	mutation probability	0.05
$g_{\mathrm{n.i.}}$	no-improvement generations to stop	10
	Trust-region algorithm	
Δ_0	initial trust-region radius	0.1
δ_+	trust-region size increase factor	2
δ_-	trust-region size decrease factor	0.5
$\Delta_{\min}$	minimum trust-region radius	$\Delta_0 \cdot \delta_-^2$
$\Delta_{\max}$	maximum trust-region radius	$\Delta_0 \cdot \delta_+^2$

Table 1. Parameter Settings for the Proposed Memetic Algorithm

All relevant parameters (e.g. initial surrogate-model sample and evolutionary parameters) were the same as in the proposed algorithm. Parameter settings are given in Table 1. To obtain statistically-significant results 30 runs were repeated for each test with the proposed algorithm and the two variants.

4.1 Mathematical test functions

For the mathematical tests functions we used the well-known Branin, Hartman 3 and Hartman 6 functions with a maximum evaluations limit of $fe_{\max}$ = 100 [13]. To asses the impact of the 'curse of dimensionality' [3] we also used the well-known chained Rosenbrock (high epistasis) and Rastrigin function (high multimodality) functions with $fe_{\max}$ = 200 [86, 90]. We set these small values for $fe_{\max}$ to measure performance under a constraint of resources (as function evaluations are considered expensive) [89]. Details of the test functions are given in Table 2. Test statistics are given in Table 3 which indicate the proposed algorithm outperformed the two surrogate-assisted EAs.

To determine in a rigorous manner if there is a statistically-significant difference between the results of the proposed algorithm and the two variants we applied the nonparametric one-tailed Mann–Whitney (or Wilcoxon) test which provides a test statistic U [35]. The null and alternative hypothesis are:

$$H_0 : P(x_i < x_p) \geqslant 0.5 \tag{25a}$$

$$H_1 : P(x_i < x_p) < 0.5\,, \tag{25b}$$

where $P(\boldsymbol{x}_i < \boldsymbol{x}_p)$ is the probability that a score of the proposed algorithm is larger (worse) than a score of one of the variants (i = 1, 2). Table 4 provides the test statistics for comparisons with the two variants over the five test functions and the decision rules. For the

Function	d	Definition	Feasible Domain	$f(\boldsymbol{x}_\mathbf{g})^1$
Branin	2	$(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	$[-5, 10] \times [0, 15]$	0
Hartman 3	3	$\sum_{i=1}^{4} c_i \exp[\sum_{j=1}^{4} a_{i,j}(x_i - p_{i,j})^2]$	$[0, 1]^d$	-3.86
Hartman 6	6	$\sum_{i=1}^{4} c_i \exp[\sum_{j=1}^{6} a_{i,j}(x_i - p_{i,j})^2]$	$[0, 1]^d$	-3.32
Rastrigin	20	$\sum_{i=1}^{n} \{x_i^2 - 10 \cdot \cos(2\pi x_i) + 10\}$	$[-5, 5]^d$	0
Rosenbrock	30	$\sum_{i=1}^{d} (2x_{i-1} - x_i^2)^2 + (1 - x_i)^2$	$[-2, 2]^d$	0

Table 2. Mathematical Test Functions

Function		Proposed	Ref.-RBF	Ref.-Kriging
	Mean	$\mathbf{9.542e-01}$	$1.324e+00$	$2.599e+00$
	S.D.	$\mathbf{6.706e-01}$	$8.142e-01$	$2.230e+00$
Branin	Median	$\mathbf{6.518e-01}$	$1.107e+00$	$1.719e+00$
	Best	$\mathbf{3.988e-01}$	$4.209e-01$	$4.268e-01$
	Worst	$\mathbf{2.593e+00}$	$3.287e+00$	$8.875e+00$
	Mean	$\mathbf{-3.764e+00}$	$-3.544e+00$	$-3.346e+00$
	S.D.	$\mathbf{1.045e-01}$	$2.157e-01$	$3.478e-01$
Hartman 3	Median	$\mathbf{-3.779e+00}$	$-3.533e+00$	$-3.426e+00$
	Best	$\mathbf{-3.862e+00}$	$-3.846e+00$	$-3.829e+00$
	Worst	$\mathbf{-3.325e+00}$	$-2.768e+00$	$-2.354e+00$
	Mean	$\mathbf{-3.014e+00}$	$-1.955e+00$	$-1.935e+00$
	S.D.	$\mathbf{2.611e-01}$	$6.159e-01$	$5.732e-01$
Hartman 6	Median	$\mathbf{-3.114e+00}$	$-1.769e+00$	$-2.020e+00$
	Best	$\mathbf{-3.259e+00}$	$-3.079e+00$	$-2.958e+00$
	Worst	$\mathbf{-2.200e+00}$	$-8.777e-01$	$-7.943e-01$
	Mean	$\mathbf{1.213e+02}$	$1.654e+02$	$1.719e+02$
	S.D.	$\mathbf{2.949e+01}$	$1.999e+01$	$2.686e+01$
Rastrigin 20	Median	$\mathbf{1.224e+02}$	$1.662e+02$	$1.693e+02$
	Best	$\mathbf{5.797e+01}$	$1.310e+02$	$1.063e+02$
	Worst	$\mathbf{1.789e+02}$	$2.173e+02$	$2.242e+02$
	Mean	$\mathbf{2.039e+01}$	$5.525e+01$	$7.333e+01$
	S.D.	$\mathbf{3.970e+00}$	$1.486e+01$	$2.508e+01$
Rosenbrock 30	Median	$\mathbf{1.988e+01}$	$5.145e+01$	$6.850e+01$
	Best	$\mathbf{1.637e+01}$	$3.324e+01$	$3.576e+01$
	Worst	$\mathbf{3.656e+01}$	$9.150e+01$	$1.418e+02$

S.D.: standard deviation

Table 3. Results for Mathematical Tests Functions

Function	Proposed-RBF	Proposed-Kriging
Branin	$2.040e{+}00$	$3.814e{+}00$
Hartman 3	$4.568e{+}00$	$5.707e{+}00$
Hartman 6	$5.914e{+}00$	$6.239e{+}00$
Rastrigin-20D	$5.219e{+}00$	$5.470e{+}00$
Rosenbrock-30D	$6.638e{+}00$	$6.623e{+}00$

Reject H_0 at $\alpha = 0.05$ if $U \geqslant 1.644$.
Reject H_0 at $\alpha = 0.01$ if $U \geqslant 2.326$.

Table 4. Mann–Whitney Test Statistics

Branin function and the RBF variant we cannot reject the null hypothesis at the 0.01 significance level, which is attributed to the relative low difficulty of the problem (d = 2) so the difference between the proposed algorithm and the variant is not statistically-significant. For all other tests we reject H_0 at both significance levels α = 0.05 and 0.01 and accept there is a statistically significant difference between the results obtained by the proposed algorithm and by each of the variants for both test functions, i.e. the proposed algorithm outperforms the two variants of the reference algorithm.

4.2 A real-world application

We have also applied the proposed algorithm to a real-world application of airfoil shape optimization. The goal is to find an airfoil shape which maximizes the lift-to-drag ratio (equivalently minimizes the drag-to-lift ratio) at the prescribed cruise conditions [51, 56], i.e.

$$\begin{aligned} &\min\ c_{\mathrm{D}}/c_{\mathrm{L}} \text{ (ratio of drag and lift coefficients)} \\ &\text{s.t. } t^{\star} = 0.12 \text{ (min. allowed thickness at 0.2–0.8 of chord)} \\ &\qquad \alpha = 2^{\circ} \text{ (cruise angle of attack)} \\ &\qquad \mathrm{M} = 0.7 \text{ (cruise Mach number)} \\ &\qquad \mathrm{h} = 30,000[\mathrm{ft}] \text{ (cruise altitude)} \end{aligned} \tag{26}$$

where the thickness constraint is based on [55] and the cruise conditions are based on [16, p.484–487] (modified from M = 0.57 , h = 25, 000[ft]) .

Accordingly, to normalize the objectives $\boldsymbol{c}_{\mathrm{D}}$ / $\boldsymbol{c}_{\mathrm{L}}$ and the thickness to the interval [0,1] we defined the objective function

$$f = \frac{c_{\mathrm{D}}}{c_{\mathrm{D,max}}} \cdot \frac{c_{\mathrm{L,min}} + \min(0.1\,,\ -1.1c_{\mathrm{L,min}})}{c_{\mathrm{L}} + \min(0.1\,,\ -1.1c_{\mathrm{L,min}})} + \frac{\max\{t^{\star} - t\,,\ 0\}}{t^{\star}}\,, \tag{27}$$

where $\boldsymbol{c}_{\mathrm{L,min}}$ = −0.5 , $\boldsymbol{c}_{\mathrm{D,max}}$ = 0.2 are the assumed minimum $\boldsymbol{c}_{\mathrm{L}}$ and maximal $\boldsymbol{c}_{\mathrm{D}}$, respectively. For the latter two only reasonable estimates are needed as they are only used to normalize the objectives.

Candidate airfoils were generated using the PARSEC parameterization [50, 76] which involves 11 design variables as shown in Figure 6. Bounds for these design variables were set according to previous studies [24, 56] and are given in Table 5. To ensure a closed airfoil shape the leading edge gap was set as Δz_{TE} = 0. Candidate airfoils were evaluated with XFoil, an analysis code for subsonic isolated airfoils based on the panel method [14]. Each

airfoil evaluation required approximately 30 seconds on a desktop computer. We set the limit of function evaluations to fe_{max} = 150.

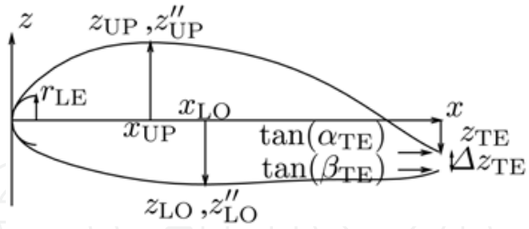


Fig. 6. PARSEC design variables.

Variable	Meaning	min.	max.
r_{LE}	leading-edge radius	0.002	0.030
x_{up}	max. upper thickness location	0.2	0.7
z_{up}	max. upper thickness	0.08	0.18
z''_{up}	max. upper curvature	−0.6	0.0
x_{lo}	max. lower thickness location	0.2	0.6
z_{lo}	max. upper thickness	−0.09	0.02
z''_{lo}	max. lower curvature	0.2	0.9
z_{TE}	trailing edge height	−0.01	0.01
Δz_{TE}	trailing edge thickness	0	0
α_{TE} [1]	upper trailing edge angle°	165	180
β_{TE} [1,2]	lower trailing edge angle°	165	190

[1] measured anti-clockwise from the x-axis.
[2] $\beta_{TE} \geqslant \alpha_{TE}$ to avoid intersecting curves.

Table 5. PARSEC design variables bounds.

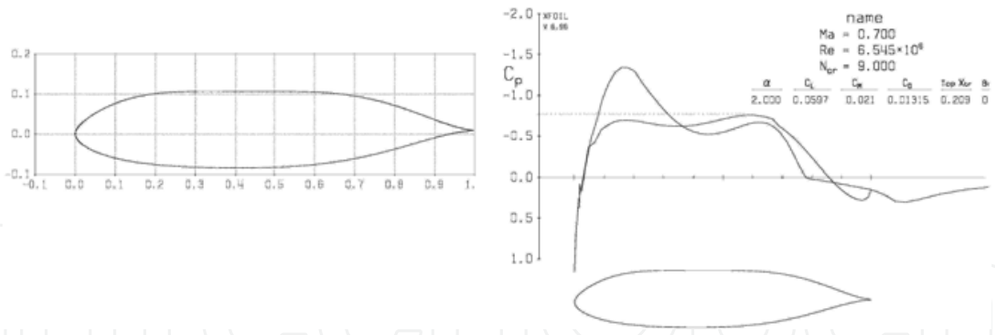
Figure 7 shows an airfoil found by the proposed algorithm and the distribution of the pressure coefficient along its upper and lower surfaces. The airfoil yields a lift to drag ratio of c_L/c_D = 4.557 and satisfies the minimum thickness requirement (minimum thickness at 0.2–0.8 of chord is t = 0.120).

We benchmarked the proposed algorithm against the two reference algorithms from the previous subsection and test statistics are given in Table 6. A nonparametric analysis similar to that of the previous section gives a Mann–Whitney test statistic of U = 3.918 and 4.110 for the RBF and Kriging variants respectively.

Statistic	Proposed	RBF	Kriging
Mean	$\mathbf{2.982e{-}01}$	$3.145e{-}01$	$3.186e{-}01$
S.D.[1]	$\mathbf{1.720e{-}02}$	$2.425e{-}02$	$4.068e{-}02$
Median	$\mathbf{2.940e{-}01}$	$3.068e{-}01$	$3.068e{-}01$
Best	$\mathbf{2.810e{-}01}$	$2.864e{-}01$	$2.885e{-}01$
Worst	$\mathbf{3.799e{-}01}$	$3.864e{-}01$	$4.699e{-}01$

[1] standard deviation

Table 6. Benchmarks for the airfoil shape optimization



(c) Airfoil geometry (d) Pressure distribution

Fig. 7. Obtained airfoil.

We thus reject the null hypothesis at both $\alpha = 0.05$ and 0.01 and accept there are statistically-significant differences between the results. This shows that also in this real-world application the proposed algorithm outperformed the surrogate-assisted variants.

5. Summary

We have proposed a surrogate-assisted memetic algorithm for expensive optimization problems. The algorithm combines global and local models and makes extensive use of model selection to assist the optimization search. The global model is an RBF artificial neural network (RBFN) whose topology is adapted incrementally to achieve both a compact network and good generalization. For the local models the proposed algorithm selects between an RBF and a Kriging model based on an accuracy assessment of the models. To ensure convergence to a true optimum of the expensive function these models are used in a trust-region framework, i.e. they replace the quadratic models; the proposed trust-region framework safeguards the accuracy of the local models and improves them, if necessary. Extensive performance analysis shows the proposed algorithm outperforms variants of a reference surrogate-assisted EA.

6. References

N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon. A trust region framework for managing use of approximation models in optimization. *Structural Optimization*, 15(1):16–23, 1998.

J. F. M. Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design—a review. *Structural optimization*, 5:129–144, 1993.

R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, N.J., 1961.

A. G. Bors and M. Gabbouj. Minimal topology for a radial basis functions neural network for pattern classification. *Digital Signal Processing*, 4:173–188, 1994.

G. E. P. Box and N. R. Draper. *Empirical Model Building and Response Surface*. John Wiley and Sons, New York, NY, 1987.

R. R. Burton. Metamodeling: A state of the art review. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, editors, *Proceedings of the 30th Conference on Winter Simulation–WSC 1998*, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

S. Chen, C. Cowan, and P. M. Grant Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309, 1991.

A. Chipperfield, P. Fleming, H. Pohlheim, and C. Fonseca. *Genetic Algorithm TOOLBOX For Use with MATLAB, Version 1.2*. Department of Automatic Control and Systems Engineering, University of Sheffield.

A. R. Conn, K. Scheinberg, and P. L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming*, 79:397–414, 1997.

P. S. de Souza and S. N. Talukdar. Genetic algorithms in asynchronous teams. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 392–397, San Mateo, Calif, 1991. Morgan Kaufmann.

J. W. Demmel. The geometry of ill-conditioning. *Computer Journal*, 3(2):201–229, 1987.

J. E. Dennis, Jr and V. Torczon. Managing approximation models in optimization. In N. M. Alexandrov and M. Y. Hussaini, editors, *Multidisciplinary Design Optimization: State of the Art*, pages 330–347. SIAM, Philadelphia, 1997.

L. C. W. Dixon and G. P. Szegö. The global optimization problem: An introduction. In L. C. W. Dixon and G. P. Szegö, editors, *Towards Global Optimisation 2*, pages 1–15. North-Holland Publishing Company, Amsterdam; New York; Oxford, 1978.

M. Drela and H. Youngren. *XFOIL 6.9 User Primer*. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2001.

D. Eby, R. C. Averill, W. F. I. Punch, and E. D. Goodman. Evaluation of injection island GA performance on flywheel design optimization. In *Proceedings of the Third Conference on Adaptive Computing in Design and Manufacturing*, pages 121–136, Plymouth, England, 1998. Springer Verlag.

A. Filippone. *Flight Performance of Fixed and Rotary Wing Aircraft*. Elsevier, first edition, 2006.

R. Franke. Scattered data interpolation: Tests of some methods. *Mathematics of Computation*, 38(157):181–200, 1982.

B. Fritzke. Fast learning with incremental RBF networks. *Neural Processing Letters*, 1(1):2–5, 1994.

A. Gaspar-Cunha and A. Vieira. A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. *International Journal of Computers, Systems and Signals*, 6(1):18–36, 2005.

K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *International Review Journal Progress in Aerospace Sciences*, 38(1):43–76, 2002.

W. E. Hart and R. K. Belew. Optimization with genetic algorithm hybrids that use local search. In R. K. Belew and M. Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, Santa Fe Institute Studies in the Sciences of Complexity, chapter 27, pages 483–496. Addison-Wesley, Reading, MA, 1995.

E. J. Hartman, J. D. Keeler, and J. M. Kowalski. Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation*, 2:210–215, 1990.

W. H. Highleyman. The design and analysis of pattern recognition experiments. *Bell Systems Technical Journal*, 41:723–744, 1962.

T. L. Holst and T. H. Pulliam. Aerodynamic shape optimization using a real-numberencoded genetic algorithm. Technical Report 2001-2473, AIAA, 2001.

R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodeling techniques under multiple modeling criteria. *Structural Optimization*, 23(1):1–13, 2001.

Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Journal of Soft Computing*, 9(1):3–12, 2005.

M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, 1990.

E. J. Kansa and Y.-C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations. *Computers and Mathematics with Applications*, 39(7):123–137, 2000.

M. K. Karakasis and K. C. Giannakoglou. On the use of surrogate evaluation models in multi-objective evolutionary algorithms. In *Proceedings of the European Conference on Computational Methods in Applied Sciences and Engineering–ECCOMAS 2004*, 2004.

M. K. Karakasis and K. C. Giannakoglou. Metamodel-assisted multi-objective evolutionary optimization. In R. Schilling, editor, *Proceedings of the 6th on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems–Eurogen 2005*, 2005.

N. B. Karayiannis and G. W. Mi. Growing radial basis neural networks:Merging supervised and unsupervised learning with network growth techniques. *IEEE Transactions on Neural Networks*, 8(6):1492–1506, 1997.

H.-S. Kim and S.-B. Cho. An efficient genetic algorithm with less fitness evaluation by clustering. In *Proceedings of 2001 IEEE Conference on Evolutionary Computation*, pages 887–894. IEEE, 2001.

K.-H. Liang, X. Yao, and C. Newton. Evolutionary search of approximated N dimensional landscapes. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 4(3):172–183, 2000.

H. Linhart and W. Zucchini. *Model Selection*. Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience Publication, New York; Chichester, 1986.

H. B. Mann and D. R. Whitney. On a test whether one of two variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18:50–60, 1947.

S. D. Marchi. On optimal center locations for radial basis function interpolation: Computational aspects. *Rendiconti del Seminario Matematico*, 61(3):343–358, 2003.

K. Marida and R. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146, 1984.

J. D. Martin and T. W. Simpson. Use of kriging models to approximate deterministic computer models. *AIAA Journal*, 43(4):853–863, 2005.

S. Matej and R. M. Lewitt. Practical considerations for 3-D image reconstruction using spherically symmetric volume elements. *IEEE Transactions on Medical Imaging*, 15(1):68–78, 1996.

C. Matheron. Principles of geostatistics. *Economic Geology*, 58:1246–1266, 1963.

M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

M. Meckesheimer, A. J. Booker, R. R. Burton, and T. W. Simpson. Computationally inexpensive metamodel assessment strategies. *AIAA Journal*, 40(10):2053–2060, 2002.

H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8:164–177, 1995.

C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.

T. J. Mitchell and M. D. Morris. Bayesian design and analysis of computer experiments: Two examples. *Statistica Sinica*, 2, 359–379 1992.

J. Moody and C. J. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm I: Continuous parameter optimization. *Evolutionary Computations*, 1(1):25– 49, 1993.

R. H. Myers and D. C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley and Sons, New York, 1995.

M. Niranjan and F. Fallside. Neural networks and radial basis functions in classifying static speech patterns. *Computer Speech and Language*, 4(3):275–289, 1990.

S. Obayashi. Airfoil shape optimization for evolutionary computation. In *Genetic Algorithms for Optimization in Aeronautics and Turbomachinery*, VKI Lecture Series 2000-07. Rhode Saint Genese, Belgium and Von Karman Institute for Fluid Dynamics, 2000.

S. Obayashi, K. Nakahashi, A. Oyama, and N. Yoshino. Design optimization of supersonic wings using evolutionary algorithms. In *Proceedings of the European Conference on Computational Methods in Applied Sciences and Engineering–ECCOMAS 1998*, pages 575–579, 1998.

Y.-S. Ong and A. J. Keane. Meta-Lamarckian learning in memetic algorithm. *IEEE Transactions On Evolutionary Computation*, 8(2):99–110, 2004.

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*, 41(4):687–696, 2003.

Y.-S. Ong, P. B. Nair, and K. Y. Lum. Max-min surrogate-assisted evolutionary algorithm for robust aerodynamic design. *IEEE Transactions on Evolutionary Computation*, 10(4):392–404, 2006.

A. Oyama, S. Obayashi, and K. Nakahashi. Real-coded adaptive range genetic algorithm and its application to aerodynamic design. *International Journal of the Japan Society of Mechanical Engineering*, 43(2):124–129, 2000.

A. Oyama, S. Obayashi, and T. Nakahashi. Real-coded adaptive range genetic algorithm applied to transonic wing optimization. In M. Schoenauer, editor, *The 6th International Conference on Parallel Problem Solving from Nature–PPSN VI*, pages 712–721, Berlin ; New York, 2002. Springer.

J. Park and I. W. Sandberg. Universal approximation using radial basis function networks. *Neural Computation*, 3:247–257, 1991.

J. Platt. A resource–allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

T. Poggio and F. Girosi. A theory of networks for approximation and learning. A. I. Memo 1140, Massachusetts Institute of Technology, Artificial Intelligence Laboratory and Center for Biological Information Processing, 1989.

T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

C. Poloni, A. Giurgevich, L. Onseti, and V. Pediroda. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design

problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 186:403–420, 2000.

D. Quagliarella and A. Vicini. Coupling genetic algorithms and gradient based optimization techniques. In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms in Engineering and Computer Science*, chapter 14, pages 288–309. John Wiley and Sons, 1997.

A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximations. In A. E. Eiben, Bäck, Thomas, M. Schoenauer, and H. -P. Schwefel, editors, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature–PPSN V*, volume 1498 of *Lecture Notes in Computer Science*, pages 87– 96, Berlin Heidelberg, 1998. Springer-Verlag.

A. Ratle. Optimal sampling strategies for learning a fitness model. In *The 1999 IEEE Congress on Evolutionary Computation–CEC 1999*, pages 2078–2085, New York, 1999. IEEE.

J.-M. Renderes and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways. In A. Sebald and L. J. Fogel, editors, *Proceedings of the Third Annual Conference on Evolutionary Programming*, pages 312–317. World Scientific, 1994.

J.-M. Renderes and S. P. Flasse. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 26(2):243–258, 1996.

S. Rippa. An algorithm for selecting a good value for the parameter *c* in radial basis function interpolation. *Advances in Computational Mathematics*, 11(2–3):193–210, 1999.

J. F. Rodríguez, J. E. Renaud, and L. T. Watson. Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data. *Structural Optimization*, 15(3/4):141–156, 1998.

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

R. Schaback. Multivariate interpolation and approximation by translates of a basis function. In C. Chui and L. Schumaker, editors, *Approximation Theory VIII*, pages 491–514. Vanderbilt University Press, 1996.

M. Sefrioui and J. Périaux. A hierarchical genetic algorithm using multiple models for optimization. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. M. Guervós, and H.-P. Schwefel, editors, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature–PPSN VI*, number 1917 in Lecture Notes in Computer Science, pages 879–888. Springer-Verlag, 2000.

T. W. Simpson, A. J. Booker, D. Ghosh, A. Giunta, P. Koch, and R.-J. Yang. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Structural and Multidisciplinary Optimization*, 27:302–313, 2004.

T. W. Simpson, D. K. J. Lin, and W. Chen. Sampling strategies for computer experiments: Design and analysis. *International Journal of Reliability and Applications*, 2(3):209–240, 2001.

T.W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen. Metamodels for computer based engineering design: Survey and recommendations. *Engineering with Computers*, 17:129–150, 2001.

R. E. Smith, B. Dike, and S. Stegmann. Fitness inheritance in genetic algorithms. In K. M. George, editor, *Proceedings of the 1995 ACM Symposium on Applied Computing–SAC 95*, pages 345–350. ACM Press, 1995.

H. Sobieckzy. Parametric airfoils and wings. In K. Fujii and G. S. Dulikravich, editors, *Recent Development of Aerodynamic Design Methodologies-Inverse Design and Optimization*, volume 68 of *Notes on Numerical Fluid Mechanics*, pages 71–88. Vieweg Verlag, Germany, 1999.

J. Sobieszczansk-Sobieski and R. Haftka. Multidisciplinary aerospace design optimization: Survey of recent developments. *Structural Optimization*, 14(1):1–23, 1997.

L. N. Søren, H. B. Nielsen, and J. Søndergaard. DACE: A MATLAB Kriging toolbox. Technical Report IMM-TR-2002-12, Informatik and Mathematical Modelling, Technical University of Denmark, 2002.

C. J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10(4):1040–1053, 1982.

M. Stone. Cross-validatory choice and assessment of statistical predictors. *Journal of the Royal Statistical Society Series B (Methodological)*, 36(2):111–147, 1974.

L. Sukhan and R. M. Kil. A Gaussian potential function network with hierarchically self-organizing learning. *Neural Networks*, 4:207–224, 1991.

Y. Tenne. Metamodel accuracy assessment in evolutionary optimization. In *Proceedings of the IEEE World Congress on Computational Intelligence–WCCI 2008*. IEEE, 2008. 1505-1512, 2008. IEEE.

Y. Tenne and S. W. Armfield. A memetic algorithm using a trust-region derivative-free optimization with quadratic modelling for optimization of expensive and noisy blackbox functions. In S. Yang, Y.-S. Ong, and Y. Jin, editors, *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51 of *Studies in Computational Intelligence*, pages 389–415. Springer-Verlag, 2007.

Y. Tenne and S. W. Armfield. A versatile surrogate-assisted memetic algorithm for optimization of computationally expensive functions and its engineering applications. In A. Yang, Y. Shan, and L. Thu Bui, editors, *Success in Evolutionary Computation*, volume 92/2008, pages 43–72. Springer, Berlin; Heidelberg, 2008.

Y. Tenne, S. Obayashi, and S. W. Armfield. Airfoil shape optimization using an algorithm for minimization of expensive and discontinuous black-box functions. In *Proceedings of the AIAA InfoTec 2007*, number AIAA-2007-2874. American Institute for Aeronautics and Astronautics (AIAA), 2007.

P. L. Toint. Some numerical results using a sparse matrix updating formula in unconstrained optimization. *Mathematics of Computation*, 32(143):839–851, 1978.

V. Torczon and M. W. Trosset. Using approximations to accelerate engineering design optimization. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 738–748, Reston, Va, 1998. American Institute for Aeronautics and Astronautics (AIAA).

A. Törn. Global optimization as a combination of global and local search. In *Proceedings of Computer Simulation Versus Analytical Solutions for Business and Economic Models*, number 17 in BAS Business Administration Studies, pages 191–206. School of Business Administration, Gothenburg, 1973.

A. Törn, M. M. Ali, and S. Viitanen. Stochastic global optimization: Problems classes and solution techniques. *Journal of Global Optimization*, 14:437–447, 1999.

A. Törn and A. Žilinskas. *Global Optimization*. Number 350 in Lecture Notes In Computer Science. Springer-Verlag, Berlin; Heidelberg; New York; London, 1989.

J. Yen, J. C. Liao, B. Lee, and D. Randolph. A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 28(2):173–191, 1998.

Y. Yun, M. Gen, and S. Seo. Various hybrid methods based on genetic algorithm with fuzzy logic controller. *Journal of Intelligent Manufacturing*, 14:401–419, 2003.

Z. Zhou, Y.-S. Ong, M. Lim, and B. Lee. Memetic algorithms using multi-surrogates for computationally expensive optimization problems. *Journal of Soft Computing*, 11(10):957–971, 2007.

Z. Zhou, Y.-S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum. Combining global and local surrogate models to accelerate evolutionary optimization. *IEEE Transactions On Systems, Man and Cybernetics-Part C*, 37(1):66–76, 2007.

Advances in Evolutionary Algorithms

Edited by Xiong Zhihui

ISBN 978-953-7619-11-4

Hard cover, 284 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

With the recent trends towards massive data sets and significant computational power, combined with evolutionary algorithmic advances evolutionary computation is becoming much more relevant to practice. Aim of the book is to present recent improvements, innovative ideas and concepts in a part of a huge EA field.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen