ENSEMBLE TOPOLOGY SELECTION IN SIMULATION-DRIVEN OPTIMIZATION

YOEL TENNE

Department of Mechanical Engineering, Ariel University Israel
E-mail: y.tenne@ariel.ac.il

Abstract - Simulation-driven optimization problems are often computationally expensive and therefore metamodels are used to obtain predicted values at a lower computational cost. To further improve the prediction accuracy ensembles combine the prediction from multiple metamodels into a single output. However, the optimal ensemble topology, namely, which metamodel variants it should incorporate, is typically not known a-priori whereas using an unsuitable topology can degrade the prediction accuracy. To address these challenges this paper proposes an algorithm which continuously adapts the ensemble topology during the search such that an optimal topology is continuously being used. Numerical tests based on a variety of test problems show the effectiveness of the proposed algorithm.

Index Terms - Black-box functions, design optimization, ensembles, metamodels.

I. INTRODUCTION

High performance computing allows engineers and scientists to improve the efficiency of the design process by evalkuating candidate designs with computer simulation instead of laboratory experiments. Such simulations, which still need to be properly validated by field experiments, transform the design process into an optimization problem with the following distinct features [33]:

•The computer simulation acts as an objective function since it assigns merit values to candidate designs. However since there if often no analytic expression which defines how a candidate design (input vector) is mapped to the simulation output the objective function is a 'black-box' function. The lack of an analytic expression presents an optimization challenge.

• Each execution of the simulation is often computationally expensiveand therefore only a small number of designs can be evaluated.

• Both the real-world physics of the problem being studied and the numerical simulation process itself can yield a function with complicated features such as multiple optima or discontinuities, which further complicates the optimization process.

An established methodology to address these challenges is to use a metamodel, namely, a computationally cheaper approximation of the true expensive function [33]. A variety of metamodels have been proposed but the optimal type is problem-dependent and is typically not known a-priori. To circumvent this ensemblesuse multiple metamodel variants and combine their prediction into a single aggregated output [12, 18, 19]. However, the effectiveness of ensembles depends on their topology, namely, which metamodel variants they incorporate but as before the optimal topology is typically unknown. To address this issue this paper proposes an optimization algorithm in which the ensemble topology is continuously being selected throughout the search such that an optimal topology is being used. To ensure convergence to a valid solution in face of inherent metamodel inaccuracies the proposed algorithm operates within a Trust Region (TR) framework. Performance analysis based on a variety of established test problems shows the effectiveness of the proposed algorithm and the merit of ensemble selection in engineering design optimization.

The remainder of this paper is organized as follows: Section II provides the pertinent background information, Section III describes in detail the proposed algorithm, and Section IV provides a detailed performance study. Lastly, Section V concludes this paper

II. BACKGROUND

As mentioned in Section I, metamodels (also termed in the literature as response surfaces or surrogates) are used as computationally cheaper approximations of the true expensive function. Metamodels are trained by using previously evaluated vectors and examples of variants include Artificial Neural Networks (ANNs), Kriging, polynomials, and radial basis functions (RBF) [19, 30]. A baseline layout for a metamodel-assisted optimization algorithm is a loop in which a metamodel is trained by using the vectors which have been evaluated so farand then a search is executed to locate a new and hopefully better solution based on the metamodel objective values. The solution vector found and possibly additional ones are evaluated with the true expensive function and the loop repeats until some termination condition is met, for example the computational budget has been exhausted. Algorithm 1 gives a pseudocode of a typical metamodel-assisted algorithm, while more involved frameworks have also been proposed [24, 31].

While metamodels offer significant enhancements to the optimization process they also introduce new challenges:

• Prediction inaccuracy : Due to the high computational

Algorithm 1: A baseline metamodel-assisted algorithm.

```
Sample an initial set of vectors;
While termination criteria not met
   train a metamodel with the vectors sampled;
   search for an optimum of the metamodel;
   evaluate the solution found and possibly
additional vectors with the true expensive
function;
end
```

cost of the simulation runs only a small number of design vectors can be evaluated and accordingly the resultant
metamodel will be inaccurate. In severe cases this inaccuracy can lead the search to convergence to a false optimum, namely, a vector which is predicted by the metamodel to be optimal but in reality is not an optimum of the true expensive function. Therefore the metamodel needs to be managed to ensure convergence toa valid solution [15]. This can be achieved with the established Trust Region (TR) framework [5, 22] in which the optimization proceeds through a series of trial-steps which are constrained to a region in which the metamodel is assumed to be sufficiently accurate. The TR is then updated based on the success or failure of these tentative trial steps, namely, whether the vector found indeed yields a better value of the true expensive function. A merit of the TR approach is that it assures asymptotic convergence to a valid optimum [6]. Section III gives a detailed description of the TR approach implemented in this study.

• Metamodel suitability: Different metamodel variants have been proposed but the optimal type is problem-dependent and is typically unknown prior to the optimization search [13, 31]. Metamodel ensembles address this issue by combining the predictions of different metamodel variants into a single output [18, 32]. The ensemble topology, namely, which metamodels it incorporates, it typically set a-priori and remains fixed throughout the search, partially since the evaluation cost of candidate designs limits the possibility of finding a suitable topology by trial-and-error type experiments. In practice the optimal topology is problem dependent and an unsuitable topology can degrade the prediction quality. As an example, ensembles were generated based on three metamodels: RBF, a radial basis functions neural network (RBFN), and Kriging, as shown in Table I. The prediction error of each ensemble was estimated based on the Root Mean Square Error (RMSE) value across four test functions in dimensions ranging from 5 to 30. Results show that the optimal ensemble topology, namely, that having the lowest RMSE, varied across the different tests and that no single topology optimal across all cases. The following section describes an optimization algorithm which addresses these shortcomings.

III. Proposed Algorithm

Leveraging on the discussion in Sections I and II, the proposed algorithm integrates the concepts of ensemble adaption, hybrid search, and trust-region approach, and operates in five main steps, as follows:

Table 1: Root Mean Square Error for Different Topologies.

Function	Ensemble Topology			
	R+KN	R+K	RN+K	R+RN +K
Ackley-5D	4.258e -01	3.702e -01	4.151e -01	2.967e -01
Rastirgin-10 D	1.223e +02	8.198e +01	1.312e +02	1.097e +02
Rosennbrock -20D	1.791e +06	1.666e +06	1.648e +06	1.693e +08
Schwefel2.1 3-30D	1.882e +06	2.179e +06	2.343e +06	2.079e +06

R:RBF, RN: RBF network, K: Kriging

Step 1) Initialization: The algorithm generates a set of vectors is by using the Optimal Latin Hypercube Sampling (OLHS) method to obtain a space-filling sample, as this improves the prediction accuracy of the resultant metamodels [34].

Step 2) The set of sampled vectors is split into a training and testing set, and for each of the j = 1...n candidate metamodel variants its respective RMSE is calculated based on the testing set as follows $e_j = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(m_j(x_i) - f(x_i))^2}$

where $m_j(x)$ is a metamodel trained based on the training set, and $x_i, i = 1 \dots l$ are the testing vectors.

Step 3) The sampled vectors are again re-split into training and testing sets, the metamodels are retrained, and for each k = 1...m candidate ensemble topology its RMSE prediction error is calculated as $\varepsilon_{k(x)} = \sum_{j=1}^{n} u_j \widehat{m_j}(x)$, $u_j = \frac{e_j^{-1}}{\sum_{j=1}^{n} e_j^{-1}}$ $e_{\varepsilon_k} = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(\varepsilon(x_i) - f(x_i))^2}$

where $\varepsilon_k(x)$ is the ensemble prediction of the kth topology, $\widehat{m_j}(x)$ is the prediction of the jth metamodel incorporated in this topology, and u_j is the metamodel's weight in the ensemble. Also, $x_i, i = 1 \dots l$ are the testing vectors, and e_ε is the root mean squared error (RMSE) of the ensemble prediction.

Step 4) The ensemble topology which yielded the best (lowest) RMSE is selected for the current optimization iteration. A new ensemble is then trained with the selected topology and with all the vectors evaluated, that is, without any training-testing splitting.

Step 5) A TR is defined around the current best vector (x_b) and a search is performed to locate the best vector in the TR. To effectively explore the entire TR and to also converge to an accurate solution a hybrid twostage search is performed: initially an explorative simulated annealing (SA) search is performed to explore the TR and to locate good solutions, as described in Appendix A, and are then refined by a localized SQP search. During this trial search only the ensemble is used, and no calls are made to the true expensive function.

Step 6) The best vector found in the previous step (x^*)is evaluated with the true objective function, and one of the following updates executes:

• If$f(x^*) < f(x_b)$: The trial step was successful since the new solution found is indeed better than the best so far. This indicates that the ensemble prediction is accurate, and so the TR is centred at the new solution and the TR radius is doubled.

• If $f(x^*) < f(x_b)$: and there are sufficient vectors inside the TR: The trial step failed since the solution found is not in fact better than the current best. This indicates that the ensemble prediction is too inaccurate, and since there are sufficient vectors in the TR the search failure is attributed to the TR being too large. Therefore, the TR radius is halved.

• If $f(x^*) \geq f(x_b)$and the number of vectors in the TR is deemed as too low: As above the trial step failed but now the failure is attributed to having too few sampled vectors in the TR. To remedy this a new vector is sampled in the TR. As a change from the classical TR framework, the proposed algorithm reduces the TR radius only if the number of vectors in the TR is sufficient, which is done to avoid premature convergence. This threshold value was calibrated with numerical experiments. In this study the candidate metamodels used were RBF, RBFN, and Kriging were used, which are described in Appendix B. However it is emphasized that the proposed algorithm can accommodate any other type or number metamodels. To complete this section, Algorithm 1gives the pseudocode of the proposed algorithm.

Algorithm 2: The Proposed Algorithm.

Generate an initial LHD sample and evaluate with the true expensive function;
Repeat
  Estimate the prediction accuracy of candidate metamodel by cross-validation;
  Re-split the sampled vectors and estimate the prediction accuracy of candidate ensembles by cross-validation;
  Select the most accurate topology as the active one in the current iteration;
  Perform a TR trial step to find an optimum in the TR based on the ensemble predictions;
  Based on the success or failure of the trial step update the TR and sample;
Until termination condition

IV. PERFORMANCE ANALYSIS

A. Mathematical test functions

In the first round of tests the proposed algorithm was used to an established suite of test functions from the literature [29] in dimensions ranging from 5 to 40, and which are shown in Table II. For comparison purposes the proposed algorithm was benchmarked against other four algorithms:

• V1: A variant of the proposed algorithm which uses a similar layout except that it used a single metamodel (RBF), and no ensembles. This algorithm was incorporated to identify any gains achieved by using an adaptive ensemble over using a single fixed metamodel.

• V2: A second variant of the proposed algorithm except that it used a fixed ensemble topology of RBF, RBFN, and Kriging metamodels. This algorithm was incorporated to identify any gains achieved by adapting the ensemble topology over using a fixed topology.

• Evolutionary algorithm with periodic sampling (EA–PS): A metamodel-assisted algorithm which is based on frameworks in [7, 23]. The algorithm combines a Kriging metamodel and an evolutionary algorithm (EA), and attempts to manage the metamodel by periodically evaluating a small subset of the EA population with the true objective function, thereby refreshing the metamodel. This algorithm is representative of many other metamodel-assisted algorithms in the literature.

•Expected Improvement with Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) (EI–CMA-ES): Another metamodel-assisted algorithm which uses a CMA-ES algorithm and a Kriging metamodels, and operates based on the expected improvement framework [3]. This algorithm represents more elaborate metamodel assisted frameworks.

The above algorithms were incorporated into the tests to evaluate: i) the contribution of dynamic ensemble adaptation (by comparing to the V1 and V2 algorithms), and ii) the performance of the proposed algorithm relative to existing algorithms in the literature. In each algorithm–test function combination 30 test runs were repeated so that a sufficient number of runs were available to support a valid statistical analysis. The number of evaluations of the true objective function was restricted to 200 as a tight limit on the number of simulation runs, so it would represent optimization settings used in practice. Table III gives the resultant test statistics of mean, standard deviation (SD), median, minimum (best) and maximum (worst) objective value in each scenario.

It follows from the test results that the proposed algorithm consistently performed well, as evident from it obtaining both the best mean statistic and best median in three cases, while in other cases its results were close to those of the best performing algorithm.

Also, its performance had a statistically significant advantage in 11 out of 16 comparisons which further indicates a solid performance. The proposed algorithm also performed well in terms of the SD statistic: it consistently achieved the best or near-best value, which indicates a robust behaviour.

The analysis also examined how the ensemble topology was updated, namely, if a single one was dominant or various topologies were evenly selected. To visualize this aspect, Figure 1 shows plots of the ensemble updates across the 200 function evaluations, where results were taken a run with the Rosenbrock-05 function and another with the Griewank-40 function. It follows that for the Rosenbrock-05 function various topologies were selected, while for the Griewank-40 mostly a single Kriging metamodel was selected.

Table 2:Test functions.

Function	Definition	Domain
Ackley	$-20\exp\left(-0.2\sqrt{\sum_{i=1}^{d} x_i^2/d}\right) - \exp\left(\sum_{i=1}^{d}\cos(2\pi x_i)/d\right) + 20 + e$	$[-32,32]^d$
Griewank	$\sum_{i=1}^{d}\{x^2{}_i/4000\} - \prod_{i=1}^{d}\{\cos(x_i)/\sqrt{i}\} + 1$	$[-100,100]^d$
Rastrigin	$\sum_{i=1}^{d}\{x_i^2 - 10\cos(2x_i)$	$[-5,5]^d$
Rosenbrock	$\sum_{i=1}^{d-1}\{100(x_i^2 + x_{i+1})^2 + (x_i - 1)^2\}$	$[-10,10]^d$
Schwefel 2.13	$\sum_{i=1}^{d}\{\sum_{j=1}^{d}[(a_{i,j}\sin(\alpha_j) + b_{i,j}\cos(\alpha_j)) - (a_{i,j}\sin(x_j) + b_{i,j}\cos(x_j))]\}^2$	$[-\pi,\pi]^d$
Weierstrass	$\sum_{i=1}^{d}\{\sum_{k=0}^{20} 0.5^k\cos(2\pi 3^k(x_i + 0.5))\} - d\sum_{k=0}^{20} 0.5^k\cos(\pi 3^k)$	$[-0.5,0.5]^d$

These variations further show the merit of adapting the ensemble topology on a per-case manner.



Figure 1: Topology selection in the test functions runs.

B. Engineering Test Problem

In the second round of tests the proposed algorithm was applied to an aerospace engineering problem of airfoil design. In this case it is required to find an airfoil shape which maximizes the lift and minimizes the aerodynamic friction drag at some prescribed flight conditions [9]. Candidateairfoils were represented with the method of Hicks

and Henne [14] in which an airfoil profile was defined by

$$y = y_b + \sum_{i=1}^{h} \alpha_i b_i(x),$$

Table 3: Statistics for the test functions.

		P	V1	V2	EAPS	E-C
Ros-05	Mean	1.513+01	3.354+01	1.387+02	2.074+02	3.701+02
	SD	3.947+01	7.510+01	2.792+01	1.640+02	2.320+01
	Med	2.512+00	3.535 +00	6.715 +00	1.796 +02	3.498 +02
	Min (best)	2.478 -02	1.741 +00	4.097 +00	1.368 +01	7.677 +01
	Max (worst)	1.242+02	2.397 +02	8.791 +02	5.617 +02	6.719 +02
Wei-10	Mean	6.633+00	8.945 +00	8.173 +00	3.706 +00	5.909 +00
	SD	1.982+00	1.740 +00	2.046 +00	5.593 -01	2.777 +00
	Med	6.399+00	9.031e+00	8.497e+00	3.702e+00	5.805 +00
	Min (best)	3.795+00	6.881e+00	5.247+00	2.787 +00	1.657 +00
	Max (worst)	1.058+01	1.244 +01	1.112 +01	4.627 +00	9.409 +00
Ras-20	Mean	6.495+01	6.509 +01	1.498 +02	1.223 +02	2.105 +02
	SD	3.761+01	1.715 +01	2.762 +01	1.219 +01	3.914 +01
	Median	4.814+01	6.688 +01	1.454 +02	1.230 +02	2.296 +02
	Min (best)	3.935+01	4.208 +01	1.520 +02	1.046 +02	1.395 +02
	Max (worst)	1.575+02	8.812 +01	1.994 +02	1.429 +02	2.507 +02
Gri-40	Mean	1.054+00	1281 +00	8.199 +00	1.461 +00	1.102 +00
	SD	2.953 -02	1.192 -01	1.158 +00	6.031 -02	3.032 -02
	Median	1.047+00	1.251 +00	8.075 +00	1.454 +00	1.096 +00
	Min (best)	1.019+00	1.128 +00	6.572 +00	1.387 +00	1.071 +00
	Max (worst)	1.118+00	1.4815+00	1.042 +01	1.595 +00	1.157 +00

Ros:Rosenbrock, Wei:Weierstrass, Ras: Rastrigin, Gri:Griewank.
P:Proposed algorithm, E-C: EI-CMAES

$$b_i(x) = \left[\sin\left(\pi x^{\log(0.5)/\log\left(\frac{i}{h+1}\right)}\right)\right]^4$$

where y_b is a baseline airfoil profile, taken here to be the NACA0012 symmetric airfoil, $b_i(x)$ are geometric basis functions [35], and $\alpha_i(x) \in [-0.01, 0.01]$ are weights which need to calibrated to obtain the optimal airfoil. To illustrate this layout Figure2 gives a schematic description. Two problem formulations were used: i) a low dimensional case where each of the upper and lower airfoil profiles were defined by three basis functions, thereby resulting in a total of six design variables, and ii) a high dimensional case where 10 basis were used per profile, thereby resulting in a total of 20 design variables. The lift and drag coefficients of candidate airfoils were obtained by using XFoil , a computational fluid dynamics simulation for analysis of subsonic isolated airfoils [8]. To ensure the structural integrity the minimum airfoil thickness (t) between 20% to 80% of the airfoil chord line was required to be equal or above a threshold value $t^* = 0.1$. Accordingly, the objective function used was

$$f = -\frac{c_l}{c_d} + p, \quad p = \begin{cases} \frac{t^*}{t} \cdot \left|\frac{c_l}{c_d}\right| \text{ if } t < t^* \\ 0 \text{ otherwise} \end{cases}$$

where p is a penalty for violation of the thickness constraint.

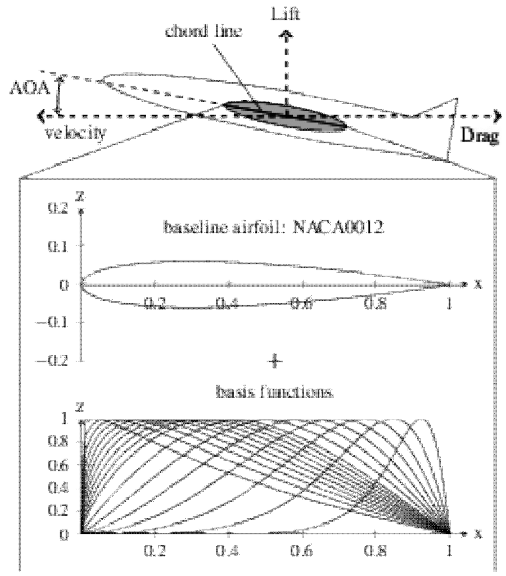


Figure 2: Layout of the airfoil problem.

The prescribed flight conditions were a cruise altitude of 30,000ft, a cruise speed of Mach 0.7, namely 70% of the speed of sound, and an angle of attack (AOA) of2°, which is the angle between the airfoil chord line and the aircraft velocity. The tests were repeated using the setup of the previous section and Table 4 gives the resultant statistics. The results follow the behaviour observed in the previous section as the proposed algorithm performed well also in these tests.

Figure 3 shows the topology updates during from two different runs, and as before the selected topology continuously varied during the search. These results, combined with the test statistics, further emphasize the merit of the proposed topology adaptation approach.

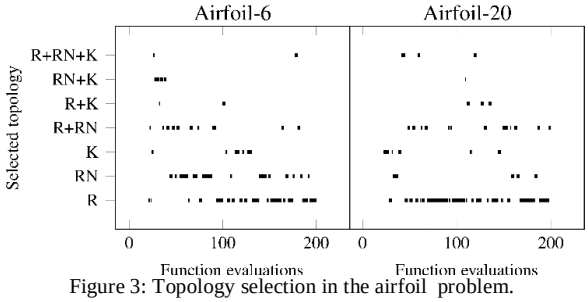


Figure 3: Topology selection in the airfoil problem.

CONCLUSION

Metamodels are used simulation-driven problems as a computationally cheaper approximations of intensive computer simulations. Given the metamodels inherent inaccuracy ensembles fuse predictions from several metamodels to improve the approximation. However, the optimal ensemble topology, that is, the exact metamodels which should be incorporated, is typically unknown a-priori. Accordinglythis study has proposed an approach in which the ensemble topology is adapted throughout the search such that optimal topology is continuously being used. To further improve the search effectiveness the proposed algorithm operates within a TR framework to ensure convergence to an optimum of the true expensive function.

Performance analysis based on both mathematical test functions and an engineering test problem show the effectiveness of the proposed algorithm and the merit of the proposed approach. Analysis also showed that the chosen ensemble topology varied throughout the search which highlights that no single topology was the overall optimal and further justifies the approach proposed.

Table 4:Statistics for the Airfoil Problems

		P	V1	V2	EA-PS	E-C
6D	Mean	-8.460+01	-8.028e+01	-8.171e+01	-7.639e+01	-7.374e+01
	SD	1.386e+01	1.633e+01	2.238e+01	2.474e+00	7.495e-01
	Med	-7.985e+01	-7.442e+01	-7.467e+01	-7.345e+01	-7.254e+01
	Min	-1.065e+02	-1.243e+02	-1.431e+02	-8.036e+01	-7.290e+01
	Max	-7.598e+01	-7.186e+01	-6.378e+01	-7.238e+01	-7.099e+01
20D	Mean	-3.249e+00	-3.198e+00	-3.241e+00	-3.174e+00	-3.212e+00
	SD	6.415e-02	6.989e-02	8.953e-02	8.887e-02	9.405e-02
	Med	-3.233e+00	-3.211e+00	-3.210e+00	-3.142e+00	-3.202e+00
	Min	-3.384e+00	-3.301e+00	-3.409e+00	-3.348e+00	-3.327e+00
	Max	-3.189e+00	-3.096e+00	-3.129e+00	-3.070e+00	-3.036e+00

Med: Median
P: Proposed algorithm, E-C:EI-CMAES

APPENDIX

The details of the metamodels used in this study are as follows:

- Kriging: This is a statistically-based metamodel which combines a global coarse approximation with a local correction which is based on the correlation between the interpolation vectors. The metamodel replicates the training sample, namely, $m(x_i) = f(x_i),\ i = 1 \dots n$

where $m(x_i)$ and $f(x_i)$ are the metamodel and true objective function, respectively, and $x_i, i = 1 \dots n$ are the sample vectors. Using a constant global function gives the Kriging metamodel

$$m(x) = \beta + \kappa(x),$$

with the constant function β and local correction $\kappa(x)$. The latter is defined by a stationary Gaussian process with mean zero and covariance

$$Cov(\kappa(x), \kappa(y)) = \sigma^2 c(\theta, x, y)$$

where $c(\theta, x, y)$ is a user-prescribed correlation function. A common choice for the latter is the Gaussian correlation function [10], defined as

$$c(\theta, x, y) = \prod_{i=1}^{d} \exp(-\theta(x_i - y_i)^2),$$

and combining it with the constant drift function transforms the metamodel from (8) into the following form

$$m(x) = \hat{\beta} + r(x)^T R^{-1}(f - 1\beta)$$

Here β is the estimated drift coefficient, R is the symmetric matrix of correlations between all interpolation vectors, f is the vector of objective values and 1 is a vector with all elements equal to 1. Also, $r(x)^T$ is the correlation vector between a new vector x and the sample vectors, namely,

$$r^T = (c(\theta, x, x_1), \dots, c(\theta, x, x_n))$$

The estimated drift coefficient $\hat{\beta}$ and variance $\hat{\sigma}^2$ are obtained from

$$\hat{\beta} = (1^T R^{-1} 1)^{-1} 1^T R^{-1} f,$$

$$\hat{\sigma}^2 = \frac{1}{n}\left[(f - 1\hat{\beta})^T R^{-1}(f - 1\hat{\beta})\right].$$

Fully defining the metamodel requires the correlation parameters θ which are commonly taken as the maximizers of the metamodel likelihood. This is achieved by minimizing the expression

$$\varphi(x) = |R|^{1/n}\hat{\sigma}^2$$

Which is a function only of θ and the sample vectors.

•Radial Basis Functions(RBF): The metamodel approximates the objective function as a superposition of basis functions of the form

$$\phi_{i(x)} = \phi(\|x - x_i\|_2)$$

where x_i is a sampled vector. Given the training sample vectors and response $x_i, f(x_i), i = 1 \dots n$ then the metamodel is given by

$$m(x) = \alpha_i \sum_{i=1}^{n} \phi_i(x) + c$$

where α_i and c are coefficients which are determined from the interpolation conditions

$$m(x_i) = f(x_i), i = 1 \dots n,$$

$$\sum_{i=1}^{n} \alpha_i = 0.$$

A common choice is the Gaussian basis function [21]

$$\phi_i(x) = \exp\left(-\frac{x - x_i}{\tau}\right)$$

where τ controls the width of the Gaussians and isdetermined by cross-validation [2, 10]. The weight coefficients α_i are obtained by solving the linear system

$$\begin{pmatrix} \Phi & 1 \\ 1^T & 0 \end{pmatrix} \alpha = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

where Φ is the Gram matrix such that $\Phi_{i,j} = \phi_i(x_j)$, 1 is a vector whose elements are all one, and f is the vector of objective values of the sample vectors.

• Radial Basis Functions Network (RBFN): This is a variant of the RBF metamodel but in which the number of basis functions is smaller than the sample size, which can improve the prediction accuracy in certain scenarios. The metamodel prediction is then given by

$$m(x) = \sum_{j=1}^{\hat{n}} \alpha_j \phi_j(x),$$

where the coefficients α_i are determined from the least-squares interpolation conditions

$$\Phi^T \Phi \alpha = \Phi^T f$$

where

$$\Phi_{i,j} = \phi_j(x_i)$$

and $x_i, i = 1 \dots n$ are the sample vectors, f is the vector of corresponding objective function values, and $\phi_j(x_i), j = 1 \dots \hat{n}$ are the basis functions which in this study were taken as the Gaussian functions described above. The basis functions centres x_j are obtained by clustering the sampled vectors and taking the resultant cluster centres.

REFERENCES

[1] A. Belloni, H. Liang, T. Narayanan, and A. Rakhlin, "Escaping the local minima via simulated annealing: Optimization of approximately convex functions," The Journal of Machine Learning Research , vol. 40, pp. 240–265, 2015.

[2] N. Benoudjit, C. Archambeau, A. Lendasse, J. Lee, and M. Verleysen, "Width optimization of the gaussian kernels in radial basis function networks," in Proceedings of the 10 th European Symposium on Artificial Neural Networks–ESANN 2002 , M. Verleysen, Ed. EvereDSide, 2002, pp. 425–432.

[3] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with Gaussian process fitness function models," IEEE Transactions on Systems, Man, and Cybernetics–Part C , vol. 35, no. 2, pp.

183–194, 2005.
[4] V. Cerny, “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm,” Journal of Optimization Theory and Applications , vol. 45, no. 1, pp. 41–51, 1985.
[5] A. R. Conn, K. Scheinberg, and P. L. Toint, “On the convergence of derivative-free methods for unconstrained optimization,” in Approximation Theory and Optimization: Tributes to M.J.D. Powell , A. Iserles and M. D. Buhmann, Eds. Cambridge; New York: Cambridge University Press, 1997, pp. 83–108.
[6] A. R. Conn, N. I. M. Gould, and P. L. Toint,Trust RegionMethods . Philadelphia, Pennsylvania: SIAM, 2000.
[7] K. A. de Jong, Evolutionary Computation:A Unified Approach . Cambridge, Massachusetts: MIT Press, 2006.
[8] M. Drela and H. Youngren, “Xfoil 6.9 user primer,” Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep., 2001.
[9] A. Filippone, Flight Performance of Fixed and Rotary Wing Aircraft , 1st ed. Amsterdam; Boston; Heidleberg: Elsevier, 2006.
[10] A. I. J. Forrester and A. J. Keane, “Recent advances in surrogate-based optimization,” Progress in Aerospace Science , vol. 45, no. 1–3, pp. 50–79, 2008.
[11] J. Frausto-Solis and F. Alonso-Pecina, “Analytically tuned parameters of simulated annealing for the timetabling problem,” WSEAS Transactions On Advances In Engineering Education , vol. 5, no. 5, pp. 272– 281, 2008.
[12] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, “Ensembles of surrogates,” Structural and Multidisciplinary Optimization , vol. 33, pp. 199–216, 2007.
[13] D. Gorissen, T. Dhaene, and F. De Turck, “Evolutionary model type selection for global surrogate modeling,” The Journal of Machine Learning Research , vol. 10, pp. 2039–2078, 2009.
[14] R. M. Hicks and P. A. Henne, “Wing design by numerical optimization,” Journal of Aircraft , vol. 15, no. 7, pp. 407–412, 1978.
[15] Y. Jin, M. Olhofer, and B. Sendhoff, “A framework for evolutionary optimization with approximate fitness functions,” IEEE Transactions on evolutionary computation , vol. 6, no. 5, pp. 481–494, 2002.
[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science , vol. 220, no. 4598, pp. 471–480, 1983.
[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and T. Edward, “Equation of state calculations by fast computing machines,” Journal of Chemical Physics , vol. 21, no. 6, pp. 1087–1092, 1953.
[18] J. Muller and R. Piché, “Mixture surrogate models based on dempster-shafer theory for global optimization problems,” Journal of Global Optimization , vol. 51, no. 1, pp. 79–104, 2011.
[19] J. Muller and C. A. Shoemaker, “Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems,” Journal of Global Optimization , vol. 60, no. 2, pp. 123–144, 2014.
[20] M.-W. Park and Y.-D. Kim, “A systematic procedure for setting parameters in simulated annealing algorithms,” Computers and Operations Research , vol. 25, no. 3, pp. 207–217, 1998.
[21] M. J. D. Powell, “Radial basis function methods for interpolation of functions of many variables,” in Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA-01) , E. A. Lipitakis, Ed. Athens, Hellas: LEA Press, 2001, pp. 2–24. [22] —, “On trust region methods for unconstrained minimization without derivatives,” Mathematical Programming, Series B , vol. 97, pp. 605–623, 2003.
[22] A. Ratle, “Optimal sampling strategies for learning a fitness model,” in The 1999 IEEE Congress on Evolutionary Computation–CEC1999 . Piscataway, New Jersey: IEEE, 1999, pp. 2078–2085.
[23] R. G. Regis and C. A. Shoemaker, “A quasi-multistart framework for global optimization of expensive functions using response surface models,” Journal of Global Optimization , vol. 56, no. 3, pp. 1719–1753, 2013.
[24] D. A. Rodríguez, P. P. Oteiza, and N. B. Brignole, “Simulated annealing optimization for hydrocarbon pipeline networks,” Journal of Industrial Engineering and Chemical Research , vol. 52, no. 25, pp. 8579–8588, 2013. [26] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiments,” Statistical Science , vol. 4, no. 4, pp. 409–435, 1989.
[25] D. J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures , 4th ed. Boca Raton, Florida: Chapman and Hall, 2007.
[26] P. Sivasankaran, T. Sornakumar, and R. Panneerselvam,“Design and comparison of simulated annealing algorithm and grasp to minimize makespan in single machine scheduling with unrelated parallel machines,” Intelligent Information Management , vol. 2, pp. 406–416, 2010.
[27] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Nanyang Technological University, Singapore and Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, Technical Report KanGAL 2005005, 2005.
[28] Y. Tenne, “A computational intelligence algorithm for simulation-driven optimization problems,” Advances in Engineering Software , vol. 47, pp. 62–71, 2012.
[29] [31] —, “An optimization algorithm employing multiple metamodels and optimizers,” International Journal of Automation and Computing , vol. 10, no. 3, pp. 227–241, 2013.
[30] —, “An algorithm for computationally expensive engineering optimization problems,” International Journal of General Systems , vol. 42, no. 5, pp. 458–488, 2013.
[31] Y. Tenne and C. K. Goh, Eds., ComputationalIntelligence in Expensive Optimization Problems , ser. Evolutionary Learning and Optimization. Berlin: Springer, 2010, vol. 2.
[32] F. A. C. Viana, G. Venter, and V. Balabanov, “An algorithm for fast optimal Latin hypercube design of experiments,” International Journal of Numerical Methods in Engineering , vol. 82, no. 2, pp. 135–156, 2009.
[33] H.-Y. Wu, S. Yang, F. Liu, and H.-M. Tsai, “Comparison of three geometric representations of airfoils for aerodynamic optimization,” in Proceedings of the 16 th AIAA Computational Fluid Dynamics Conference . Reston, Virginia: American Institute of Aeronautics and Astronautics, 2003, pp. 1–11, AIAA 2003-4095.
[34] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng, “Generalized simulated annealing for global optimization: The gensa package,” The R Journal , vol. 5, no. 1, pp. 13–28, 2013.
[35] B. Yang, L. Guang, T. Säntti, and J. Plosila, “Parameteroptimized simulated annealing for application mapping on networks-on-chip,” in Learning and Intelligent Optimization: 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers , Y. Hamadi and M. Schoenauer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 307–322.

★★★