International Journal of Control, Automation and Systems 18(4) (2020) 955-965
http://dx.doi.org/10.1007/s12555-018-0356-7

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Online Ensemble Topology Selection in Expensive Optimization Problems

Yoel Tenne

Abstract: Simulation-driven optimization problems are often computationally-expensive, an aspect which has motivated the use of metamodels as they provide approximate function values more economically. To further improve the prediction accuracy the use of ensembles has been explored in which predictions from multiple metamodels are combined. However, the optimal ensemble topology, namely, which types of metamodels it includes, is typically not known, while using a fixed topology may degrade the prediction accuracy and search effectiveness. To address this issue this paper proposes a metamodel-assisted algorithm which autonomously adapts the ensemble topology online during the search such that an optimal topology is used throughout. An extensive performance analysis shows the effectiveness of the proposed algorithm and approach.

Keywords: Expensive optimization problems, metamodels, operations research, simulations.

1. INTRODUCTION

Engineers and researchers often use computer simulations to evaluate candidate designs so that the duration and cost of the design process can be reduced. Such simulations, which still need to be validated with real-world experiments, transform the design process into an optimization problem with three distinct features [1]:

- The simulation acts as the objective function since it assigns objective values to candidate designs but an analytic expression for this function is unavailable. This might be because the output value results from intricate calculations such as finite differences methods, or because it is a commercial or a legacy code. Such a *black-box function* precludes the use of optimization algorithms which require an analytic function.
- Each simulation run is *computationally expensive*, that is, it requires a lengthy run time, and this severely restricts the number of candidate designs which can be evaluated.
- Both the real-world physics being modelled and the numerical simulation process itself often yield an objective function which has a complicated and non-convex landscape which further complicates the optimization process.

In such settings it is common to use a *metamodel*, namely, a mathematical approximation of the true expensive function which provides predicted objective values at a lower computational cost [1–3]. A variety of metamodels have been proposed but the optimal type is problem-dependant and is typically not known a-priori. *Ensembles* attempt to address this issue by aggregating predictions from several metamodels into a single one [4, 5]. However the effectiveness of ensembles depends on their topology, namely, which metamodels they incorporate, but the optimal topology is usually unknown a-priori. Furthermore, due to the high evaluation cost of the simulation it is typically impractical to identify the optimal topology in a 'trial and error' approach. To address this challenge this paper proposes an algorithm in which the ensemble topology is continuously adapted throughout the search in a rigorous and efficient way such that the most suitable ensemble is continuously being used. The proposed algorithm also operates within a trust-region (TR) framework to ensure convergence to an optimum of the true expensive objective function. An extensive performance analysis using both mathematical test functions and an engineering test problem shows the effectiveness of the proposed algorithm and highlights the merit of the proposed topology selection approach. The remainder of this paper is organized as follows: Section 2 provides the background information, Section 3 describes in detail the proposed framework and algorithm, and Section 4 describes an extensive performance analysis and discussion of the results. Lastly, Section 5 concludes this paper.

2. BACKGROUND

As mentioned in Section 1 expensive optimization problems are prevalent and Fig. 1 gives a schematic layout. In such cases metamodels are often used to approximate the simulation's input-output relation and to pro-

Manuscript received May 28, 2018; revised May 18, 2019; accepted August 22, 2019. Recommended by Associate Editor Tae-Hyoung Kim under the direction of Editor Euntai Kim.

Yoel Tenne is with the Department of Mechanical and Mechatronic Engineering, Ariel University, Israel (e-mail: y.tenne@post.com).

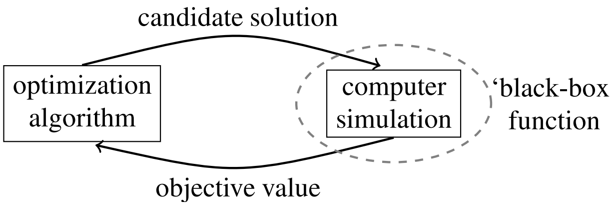

Fig. 1. The layout of an expensive black-box optimization problem.

vide approximate objective values at a lower computational cost. Metamodels are typically interpolants which have been trained with previously evaluated vectors and examples include artificial neural networks (ANNs), Kriging, and radial basis functions (RBF) [6–9].

However, the use of metamodels introduces new challenges into the optimization search:

- Prediction inaccuracy: Since the computer simulation is resource-intensive only a small number of vectors can be evaluated and used to train the metamodel. This in turn leads to an inaccurate metamodel which adversely impacts the search [10]. It is therefore necessary to *manage* the metamodel to ensure its accuracy during the search. To accomplish this the proposed algorithm leverages on the *trust-region* (TR) approach from the field of nonlinear programming [11, 12]. In this approach the optimization is preformed through a sequence of trial steps which are constrained to a TR, namely, the region where the metamodel is assumed to be accurate. Based on the success of the trial steps, namely, if a new optimum was indeed found, the TR is updated. Section 3 describes the TR approach implemented in this study.
- Metamodel suitability: Various metamodels have been proposed but the most suitable variant is problem dependant and is typically unknown a-priori [13–15]. Ensembles aim to address this by aggregating the predictions of multiple metamodels into a single one to improve the overall prediction accuracy [4, 5]. Fig. 2 shows an example of an ensemble setup based on the Rosenbrock function. The ensemble *topology*, namely, the set of metamodels it incorporates, is typically chosen a-priori and is unchanged during the search. However, this can harm the search since an unsuitable topology degrades the prediction accuracy. As an example, radial basis functions (RBF), radial basis functions neural network (RBFN), and Kriging metamodels were used to generate four different ensembles and their accuracies were measured with the root mean square error (RMSE) across four test functions in dimensions 5 to 30. Table 1 gives the test results from which it follows that the prediction accuracy varied with topology and that no single topology was optimal. To address this issue an optimization algorithm is proposed in which an optimal topology is continuously being selected during the search, as described in the following section.

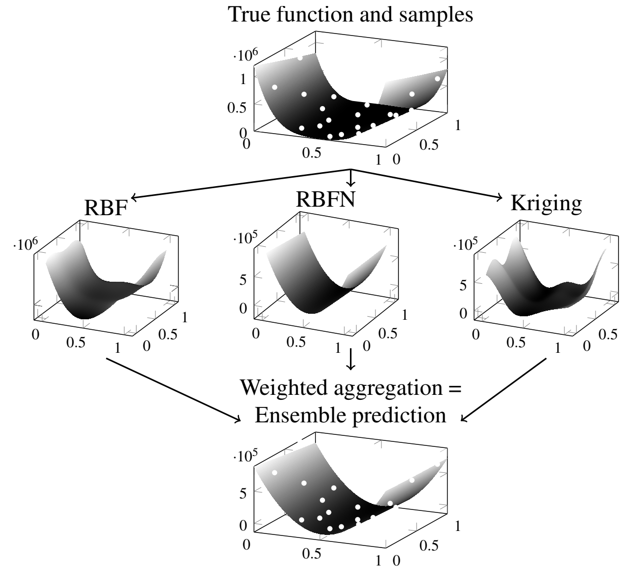

Fig. 2. An ensemble generated based on a sample of 20 vectors of the Rosenbrock function. The metamodels used are radial basis function (RBF), RBFN, and Kriging.

Table 1. RMS prediction error by topology.

	Ensemble Topology			
Function	R+N	R+K	N+K	R+N+K
Ack-5D	4.258e-01	3.702e-01	4.151e-01	**2.967e-01**
Ras-10D	1.223e+02	**8.198e+01**	1.312e+02	1.097e+02
Ros-20D	1.791e+06	1.666e+06	**1.648e+06**	1.693e+06
Sch-30D	**1.882e+06**	2.179e+06	2.343e+06	2.079e+06

Best entry per row is emphasized.
R: RBF, N: RBF network, K: Kriging.
Ack: Ackley, Ras: Rastrigin, Ros: Rosenbrock,
Sch: Schwefel 2.13.

3. PROPOSED ALGORITHM

To address the issue of inadequate ensemble topology an algorithm with online topology selection is proposed. The algorithm operates in five main steps, as follows:

1) Initialization: An initial set of vectors is sampled to enable training the metamodels. In this study the orthogonal Latin hypercube design (OLHD) sampling method was used to ensure that an adequate space-filling sample is obtained which in turn improves the accuracy of the resultant metamodels [16, 17].
2) The vectors which have been sampled are split into a training and testing set and the accuracy of each candidate metamodel is estimated with its root mean square

error (RMSE) of prediction, namely

$$e_j = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\left(m_j(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\right)^2}\,, \tag{1}$$

where $m_j(\boldsymbol{x})$ is the metamodel generated based on the training set $\boldsymbol{x}_i, i = 1\ldots l$, and $f(\boldsymbol{x})$ is the true objective function response. The split ratio between the training and testing sets in this step was calibrated through numerical experiments as described in Section 4.1.

3) The sampled vectors are now re-split again into training and testing sets and each metamodel variant is retrained. Next, for each candidate ensemble topology its overall prediction and metamodel weights are calculated as

$$\hat{f}_k(\boldsymbol{x}) = \sum_{j=1}^{n_k} u_j \hat{m}_j(\boldsymbol{x}), \quad u_j = \frac{e_j^{-1}}{\sum_{j=1}^{n_k} e_j^{-1}}, \tag{2a}$$

where $\hat{f}_k(\boldsymbol{x})$ is the overall ensemble prediction and $\hat{m}_j(\boldsymbol{x}),\ j = 1\ldots n_k$ are the participating metamodels. The weight of each metamodel is determined such that it is inversely proportional to the metamodel RMSE error obtained in eq. (1). Lastly the prediction accuracy of the candidate ensemble is estimated as

$$\varepsilon_k = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\left(\hat{f}_k(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\right)^2}, \tag{2b}$$

where $\boldsymbol{x}_i, i = 1\ldots l$ are the testing vectors in the current testing set. The split ratio between the training and testing sets in this step was calibrated through numerical experiments as described in Section 4.1.

4) The ensemble topology which achieved the lowest RMSE is selected as the best topology and a corresponding new ensemble is trained by using all the vectors stored in memory.

5) A trust-region (TR) search is performed to find a new and improved solution vector. First, a TR is defined around the current best vector ($\boldsymbol{x}_b$) as a sphere of radius Δ, namely

$$\mathcal{T} : \|\boldsymbol{x} - \boldsymbol{x}_b\|_2 \leqslant \Delta\,. \tag{3}$$

Next a search if performed to locate an optimum in the TR, and for efficiency a hybrid search is used which combines an evolutionary algorithm (EA) and an SQP algorithm such that the EA explores the domain globally while the SQP converges locally. During the search responses are obtained only from the ensemble without any calls to the true expensive function.

6) The new vector found ($\boldsymbol{x}^\star$) in the TR search is evaluated with the true expensive function and is compared to $\boldsymbol{x}_b$. Based on this comparison and assuming a minimization problem then one of the following updates take place:

- If $f(\boldsymbol{x}^\star) < f(\boldsymbol{x}_b)$: The trial step is considered successful since the vector found is indeed better than the current best one. This implies that the ensemble is accurate and so the TR radius is doubled to support a search in a larger region.
- If $f(\boldsymbol{x}^\star) \geqslant f(\boldsymbol{x}_b)$ *and* there is a *sufficient* number of vectors in the TR: The trial step was unsuccessful since the vector found was not in fact better than the current best one. This implies that the ensemble prediction is inaccurate but since the number of vectors in the TR is deemed as sufficient the poor prediction accuracy is attributed to the TR being too large for the ensemble to be accurate in. Accordingly, the TR radius is halved.
- If $f(\boldsymbol{x}^\star) \geqslant f(\boldsymbol{x}_b)$ *and* there is an *insufficient* number of vectors in the TR: as above the trial step was unsuccessful but now the poor prediction accuracy is attributed to having too few vectors in the TR. Accordingly, new vectors are then sampled in the TR.

As a change from the classical TR framework the proposed algorithm contracts the TR only if the number of vectors in the TR is above a threshold value to avoid premature convergence. This threshold was calibrated through numerical experiments as detailed in Section 4.1.

Another change from the classical framework is the addition of new vectors in the TR to improve the metamodels prediction accuracy. The latter can be improved locally around the current optimum or globally in regions sparse with vectors. To achieve these opposing goals the vectors are obtained by minimizing the function

$$h(\boldsymbol{x}) = w h_1(\boldsymbol{x}) + (1 - w) h_2(\boldsymbol{x})\,, \tag{4}$$

with an EA. Here $h_1(\boldsymbol{x})$ is a rank assigned based on the objective value with 1 assigned to the best, and $h_2(\boldsymbol{x})$ is a rank based on the distance from existing vectors in the TR with 1 assigned to the farthest vector. The weight w represents a trade-off between the two objectives. Ranks are used to make the process insensitive to the magnitude and sign of the objective values and the distances. The number of vectors sampled and their weights were calibrated by numerical experiments as detailed in Section 4.1.

It is emphasized that while in this study the RBF, RBFN, and Kriging metamodels were used the proposed algorithm can accommodate any other type or number of metamodels. To complete the description Fig. 1 gives the full pseudocode.

4. PERFORMANCE ANALYSIS

4.1. Parameter calibration

As described in Section 3 the proposed algorithm relies on four main parameters: i) the threshold number of

Algorithm 1: Proposed algorithm with online topology selection.

generate an initial sample and evaluate with the true expensive function;
repeat
  split the cached vectors into training-testing sets and estimate the accuracy of candidate metamodels;
  re-split the cached vectors and estimate the accuracies of the candidate topologies;
  train an ensemble based on the most accurate topology found;
  perform a TR trial step and update the sample;
  add the new vectors sampled to the cache;
until *maximum number of analyses completed*;

TR vectors needed for TR contraction, ii) the number and weights of new vectors generated in the TR, iii) the split ratio in the metamodels evaluation, and iv) the split ratio in the topologies evaluation.

The above parameters were calibrated based on numerical experiments in which one parameters was modified in turn while all others were kept at prescribed baseline values. For each candidate parameter setting the algorithm was run ten times with the functions Rastrigin and Rosenbrock each in 10D and 20D respectively, resulting in 120 runs per (3 candidate values × 4 objective functions × 10 repetitions). Table 2 gives the tests statistics and the ranks for each candidate settings where lower ranks indicate a better performance. The functions are abbreviated in the table by Ras for Rastrigin and Ros for Rosenbrock. Based on the results the following settings were chosen: i) threshold number of TR vectors: there was no conclusive optimal value hence the middle setting of $0.5d$ was chosen, ii) new TR vectors: sampling four new vectors with the weights $w = \{0.8, 0.2\}$, iii) training-testing split ratio for metamodels: 40%-60%, iv) training-testing split ratio for ensembles: 60%-40%. The chosen settings are emphasized in bold for each evaluation.

Table 2. Results for the parameter sensitivity analysis.

(a) Threshold number of vectors for TR contraction.

Function	$0.1d$		$0.5d$		d	
	Mean	R	Mean	R	Mean	R
Ras-10	4.598e+01	02	4.827e+01	03	3.530e+01	01
Ros-10	1.233e+02	02	4.393e+01	01	1.317e+02	03
Ras-20	9.256e+01	03	8.656e+01	02	7.997e+01	01
Ros-20	4.503e+02	01	5.361e+02	02	8.518e+02	03
Overall		08		**08**		08

d: Dimension of objective function.

(b) Weights for generating new TR vectors.

Function	$\{0.8, 0.6, 0, 4, 0.2\}$		$\{0.8, 0.5, 0.2\}$		$\{0.8, 0.2\}$	
	Mean	R	Mean	R	Mean	R
Ras-10	4.598e+01	03	4.021e+01	02	3.796e+01	01
Ros-10	1.233e+02	03	7.385e+01	02	6.745e+01	01
Ras-20	9.256e+01	03	7.364e+01	02	7.084e+01	01
Ros-20	4.503e+02	01	5.618e+02	03	4.794e+02	02
Overall		10		09		**05**

(c) Split ratio for metamodel assessment.

Function	80-20%		60-40%		40-60%	
	Mean	Rank	Mean	Rank	Mean	Rank
Ras-10	4.598e+01	02	3.981e+01	01	5.414e+01	03
Ros-10	1.233e+02	02	1.266e+02	03	4.929e+01	01
Ras-20	9.256e+01	03	8.899e+01	02	7.970e+01	01
Ros-20	4.503e+02	01	5.578e+02	03	5.318e+02	02
Overall		08		09		**07**

(d) Split ratio for ensemble assessment.

Function	80-20%		60-40%		40-60%	
	Mean	Rank	Mean	Rank	Mean	Rank
Ras-10	4.598e+01	02	4.240e+01	01	4.779e+01	03
Ros-10	1.233e+02	02	1.441e+02	03	7.509e+01	01
Ras-20	9.256e+01	03	6.778e+01	01	8.337e+01	02
Ros-20	4.503e+02	01	5.379e+02	02	8.002e+02	03
Overall		08		**07**		09

4.2. Mathematical test functions

For a rigorous performance evaluation the set of test functions from [18] was used in dimensions 5-40, and Table 3 gives their details. Additionally four competing algorithms were also incorporated in the tests to assess the relative performance of the proposed algorithm:

- V1: A variant of the proposed algorithm which is identical in operation except that it used a single metamodel (RBF) but no ensemble topology selection.
- V2: A variant of the proposed algorithm which is identical in operation except that it used a single fixed ensemble consisting of RBF, RBFN, and Kriging metamodels.
- EA with periodic sampling (EA–PS) [19]: The algorithm combines a Kriging metamodel and an EA, and safeguards the metamodel accuracy by periodically evaluating a small subset of the population with the true objective function and incorporating them into the metamodel.
- Expected Improvement with covariance matrix adaption evolutionary strategies (EI–CMA-ES) [20]: The algorithm combines a covariance matrix adaptation evolutionary strategy (CMA-ES) algorithm with Kriging metamodels and uses the expected improvement framework to update the metamodels.

Table 3. Mathematical test functions.

Fun.	Definition	Domain
Ack	$-20\exp\left(-0.2\sqrt{\frac{x_i^2}{d}}\right) - \exp\left(\frac{\sum_{i=1}^{d}\cos(2\pi x_i)}{d}\right)+20+e$	$[-32,32]^d$
Gri	$\sum_{i=1}^{d}\{x_i^2/4000\} - \prod_{i=1}^{d}\{\cos\left(x_i/\sqrt{i}\right)\} + 1$	$[-100,100]^d$
Ras	$\sum_{i=1}^{d}\left\{x_i^2 - 10\cos(2\pi x_i)+10\right\}$	$[-5,5]^d$
Ros	$\sum_{i=1}^{d-1}\left\{100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right\}$	$[-10,10]^d$
Sch	$\sum_{i=1}^{d}\{\sum_{j=1}^{d}\left[(a_{i,j}\sin(\alpha_j)+b_{i,j}\cos(\alpha_j)) - (a_{i,j}\sin(x_j)+b_{i,j}\cos(x_j))\right]\}^2$	$[-\pi,\pi]^d$
Wei	$\sum_{i=1}^{d}\{\sum_{k=0}^{20}0.5^k\cos\left(2\pi 3^k(x_i+0.5)\right)\} - d\sum_{k=0}^{20}0.5^k\cos(\pi 3^k)$	$[-0.5,0.5]^d$

Ack:Ackley, Gri:Griewank, Ras:Rastrigin
Ros:Rosenbrock, Sch:Schwefel 3.12, Wei:Weierstrass

This test setup was implemented to benchmark the proposed algorithm against variants with no ensemble or ensemble selection and against reference algorithms from the literature. In each algorithm-test function combination 30 runs were repeated to support a valid statistical analysis while the limit of evaluations of the true objective function was 200 to represent real-world settings. Since these are mathematical test functions which do not require a simulation-code run the evaluation time per candidate vector was negligible, while the total duration of an optimization run time for the full 200 evaluations was in the range of from 2-5 minutes.

Table 4 gives the resultant test statistics and the statistical significance level α of the performance gains and calculated with the Mann–Whitney nonparametric test [21].

Test results show that the proposed algorithm performed well as it achieved the best mean and median statistics in nine out of twelve cases. Analysis of the standard deviation (SD) statistic shows that there was some variation in the performance of proposed algorithm across the test functions. It achieved the best (lowest) SD for the Rosenbrock-20D and Schwefel 2.13-40D functions, and was the third to fifth best in the remaining functions. The test results also show that the online topology selection was superior to using no ensembles at all by comparison to the V1 algorithm, and was also superior to using a fixed ensemble topology by comparison to the V2 algorithm. The results also show that the proposed algorithm outperformed two representative algorithms from the literature. Lastly, the proposed algorithm had a statistically significant performance advantage over the reference algorithms in 28 out of 48 comparisons (four reference algorithms $\times$ twelve test cases), which further demonstrates its effectiveness.

The impact of the number of metamodels being used on run-time duration was also investigated. To thoroughly address this issue the required execution time of the proposed method was studied across various test functions and dimensions, and in scenarios involving 1 or 3 metamodels, respectively. Table 5 gives the resultant statistics from which it follows that in five out of the eight cases the change was less than 5% while only for Rastrigin-05, Schwefel-05, and Weierstrass-40 the change was in the range of 14%-17%. Overall these results show there is only a minor and often negligible increase in the run-time duration as the number metamodels used increases which highlights the effectiveness of the proposed algorithm.

An analysis was also performed on the pattern of topology updates to study if one topology was predominately selected or varied topologies were used. Accordingly Fig. 3 shows the topologies selected in one run with the Ackley-10D function and one with the Rosenbrock-20D function. It follows that in both cases different topologies were selected. While a topology consisting of only a Kriging metamodel was selected more frequently no single topology was consistently optimal. These results further highlight the merit of the proposed approach of online topology selection.

Also, the convergence trends of the five algorithms are compared in Fig. 4 based on runs with the Ackley-10D and Rosenbrock-20D functions. It follows that the proposed algorithm was among the fastest to converge and achieved the best final result. The contribution of the ensemble topology selection is evident from comparisons to the V1 and V2 variants which converged more slowly and achieved inferior final results.

For completeness the analysis also examined if the performance gains of the proposed algorithm diminished or even disappeared in certain settings. Accordingly, Table 6 gives the percentage difference between the mean results of the proposed algorithm and the reference algorithms. From analyzing these the following can be observed:

- The proposed algorithm consistently outperformed the reference algorithms when the objective functions had a prominent convex trend, namely, Ackley,

Table 5. Mean run-time by number of metamodels.

Function	1 Metamodel	3 Metamodels	% change
Rastrigin-05	23.8	27.4	15.1
Rastrigin-20	20.6	20.8	0.9
Rosenbrock-05	19.4	18.4	-5.1
Rosenbrock-20	19.2	20.0	4.1
Schwefel-05	19.4	22.2	14.4
Schwefel-40	29.6	28.4	-4.0
Weierstrass-10	20.6	20.2	-1.9
Weierstrass-40	23.6	27.8	17.8

Results are in minutes.

Table 4. Statistics for the test function problems.

		P	V1	V2	EAPS	EICMA
Ack10	Mean	**7.705+00**	1.455+01	1.356+01	5.241+00	1.796+01
	SD	8.359+00	4.649+00	8.051+00	5.590-01	1.529+00
	Median	**2.314+00**	1.592+01	1.908+01	5.408+00	1.797+01
	Min	9.007-02	2.383+00	3.457+00	4.098+00	1.443+01
	Max	1.836+01	1.825+01	2.048+01	6.010+00	1.988+01
	α			0.01		0.01
Gri10	Mean	**1.304-01**	1.972-01	2.078-01	9.579-01	9.338-01
	SD	1.851-01	1.714-01	2.213-01	1.076-01	2.435-01
	Median	**7.747-02**	1.294-01	1.357-01	9.862-01	1.007+00
	Min	9.350-03	3.569-02	2.290-02	7.146-01	2.441-01
	Max	6.505-01	5.661-01	7.601-01	1.046+00	1.050+00
	α				0.01	0.01
Ras05	Mean	**6.377+00**	9.360+00	8.018+00	7.631+00	2.131+01
	SD	3.728+00	7.852+00	8.349+00	4.811+00	4.890+00
	Median	5.980+00	7.464+00	**4.298+00**	7.226+00	2.139+01
	Min	1.997+00	1.005+00	3.369+00	1.621+00	1.353+01
	Max	1.195+01	2.787+01	3.076+01	1.456+01	3.006+01
	α					0.01
Ros05	Mean	**1.477+01**	3.317+01	1.369+02	2.074+02	3.701+02
	SD	3.931+01	7.649+01	2.801+02	1.640+02	2.320+02
	Median	**2.578+00**	3.616+00	6.765+00	1.796+02	3.498+02
	Min	2.528-02	1.783+00	4.277+00	1.368+01	7.677+01
	Max	1.265+02	2.443+02	8.842+02	5.617+02	6.719+02
	α			0.01	0.01	0.01
Sch05	Mean	5.438+02	3.973+02	3.710+02	5.598+02	**3.333+02**
	SD	8.945+02	9.715+02	6.578+02	4.995+02	3.227+02
	Median	7.454+01	**2.814+00**	1.763+02	4.804+02	2.050+02
	Min	2.614-02	6.026-02	5.037+01	5.685+01	3.426+01
	Max	2.205+03	3.088+03	2.221+03	1.817+03	1.080+03
	α					
Wei10	Mean	6.742+00	9.217+00	8.421+00	**4.817+00**	5.909+00
	SD	1.995+00	1.805+00	2.113+00	7.271-01	2.777+00
	Median	6.580+00	9.346+00	8.870+00	**4.813+00**	5.805+00
	Min	3.934+00	7.116+00	5.315+00	3.623+00	1.657+00
	Max	1.079+01	1.293+01	1.166+01	6.016+00	9.409+00
	α		0.01			
Ack20	Mean	**6.594+00**	9.028+00	1.998+01	6.814+00	1.863+01
	SD	5.852+00	6.503+00	3.322-01	2.461-01	1.921+00
	Median	**4.016+00**	5.718+00	2.001+01	6.744+00	1.934+01
	Min	2.772+00	3.605+00	1.928+01	6.468+00	1.493+01
	Max	1.835+01	1.875+01	2.041+01	7.203+00	2.044+01
	α		0.05	0.01	0.05	0.01
Gri40	Mean	**1.068+00**	1.307+00	8.421+00	1.461+00	1.102+00
	SD	2.697-02	1.279-01	1.113+00	6.031-02	3.032-02
	Median	**1.065+00**	1.292+00	8.252+00	1.454+00	1.096+00
	Min	1.037+00	1.163+00	6.722+00	1.387+00	1.071+00
	Max	1.130+00	1.502+00	1.039+01	1.595+00	1.157+00
	α		0.01	0.01	0.01	0.01
Ras20	Mean	**6.696+01**	6.708+01	9.603+01	1.223+02	2.105+02
	SD	3.865+01	1.692+01	3.047+01	1.219+01	3.914+01

	Median	**4.943+01**	6.964+01	9.286+01	1.230+02	2.296+02
	Min	3.938+01	4.392+01	4.843+01	1.046+02	1.395+02
	Max	1.605+02	8.870+01	1.530+02	1.429+02	2.507+02
	α			0.05	0.01	0.01
Ros20	Mean	**5.839+02**	1.031+03	8.186+02	8.435+02	3.967+03
	SD	2.094+02	5.818+02	3.823+02	3.012+02	9.406+02
	Median	**5.956+02**	8.665+02	7.932+02	7.782+02	3.685+03
	Min	2.143+02	5.483+02	3.078+02	4.676+02	3.141+03
	Max	8.905+02	2.517+03	1.521+03	1.439+03	6.144+03
	α		0.01		0.05	0.01
Sch40	Mean	**7.727+05**	8.981+05	1.935+06	1.774+06	1.667+06
	SD	2.219+05	2.571+05	6.789+05	2.509+05	6.520+05
	Median	**7.243+05**	8.622+05	2.032+06	1.744+06	1.528+06
	Min	5.130+05	5.885+05	8.715+05	1.415+06	8.933+05
	Max	1.131+06	1.362+06	3.065+06	2.104+06	2.871+06
	α			0.01	0.01	0.01
Wei40	Mean	**2.824+01**	4.160+01	4.394+01	3.045+01	3.598+01
	SD	4.401+00	4.261+00	3.885+00	1.645+00	1.463+01
	Median	**2.547+01**	4.227+01	4.461+01	2.995+01	2.597+01
	Min	2.421+01	3.353+01	3.726+01	2.878+01	2.100+01
	Max	3.482+01	4.794+01	4.867+01	3.337+01	5.817+01
	α		0.01	0.01		

Ros:Rosenbrock, Sch:Schwefel-3.12, Wei:Weierstrass

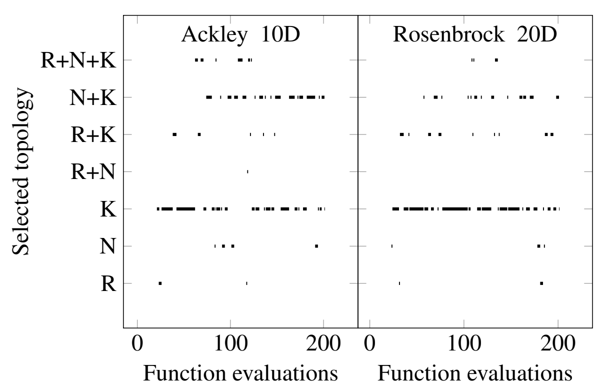

Fig. 3. Selected topologies for two test functions. R: RBF, N: RBFN, K: Kriging.

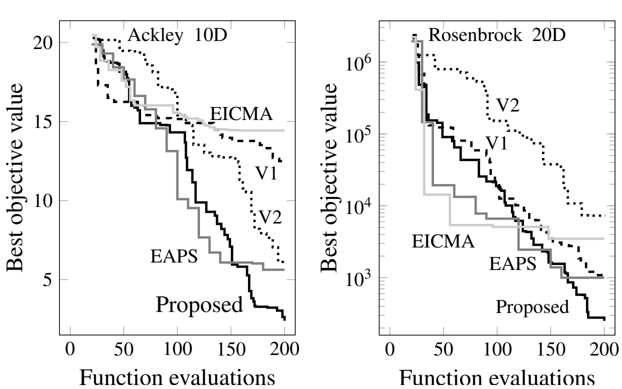

Fig. 4. Convergence plots for two test functions.

Griewank, Rosenbrock, Rastrigin, as evident from the positive percentages. However, with the Schwefel and Weierstrass functions which do not have this feature the reference algorithms were occasionally better. This implies that a prominent nonconvex landscape was more challenging to model even with an ensemble and in such scenarios the advantage of the proposed algorithm diminished. The bivariate version of these functions is shown in Fig. 5 in which the convex and nonconvex landscapes can be seen.

- The relative performance gains of the proposed algorithm did not vary much with the function dimension. This is evident from comparing the results for the same reference algorithm and objective function across different dimensions, namely, only in 11 out of 24 comparisons was the difference in the low dimensional case larger than in the corresponding high dimensional case. These results indicate that all algorithms were similarly affected by the increased dimensionality and the relative performance gains were maintained.
- Using a single metamodel versus using a topologically-complicated ensemble had little impact on the relative performance gains of the proposed algorithm. This is indicated by comparing the differences in the means between the proposed algorithm and the V1 and also the V2 variants, namely, only in 5 out of 12 comparisons the V2 (ensemble) results were lower than the V1 (single metamodel) results.

Overall the main factor which impacted the perfor-

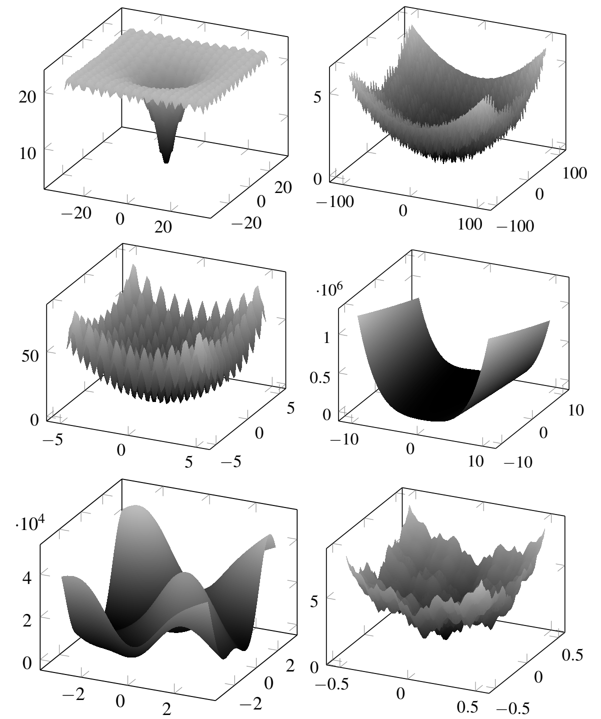

Fig. 5. The test functions used from top left horizontally: Ackley, Griewank, Rastrigin, Rosenbrock, Schwefel 3.12, Weierstrass.

mance of the proposed algorithm was the landscape of the objective function. In functions which do not have a prominent global convex shape the performance advantage of the proposed algorithm was less obvious. However, the function dimension or the complexity of the metamodels being used did not strongly affect performance.

4.3. Engineering problem

To augment the preceding analysis an additional test problem was used in which objective values were obtained from a computer simulation, as in real-world simulation-driven problems. The aerospace problem used is that of designing an airfoil which minimizes its aerodynamic friction force (also termed *drag*) while maximizing its beneficial lift force, all at some prescribed flight condition. In practice the drag and lift are represented by the drag and lift coefficients c_d and c_l, respectively. Fig. 6 gives a schematic layout of the airfoil problem. Such airfoil problems are well-established test cases for evaluating the effectiveness of optimization algorithms, for example [22, 23].

To represent candidate airfoils the method of [24] was used in which an airfoil profile is defined as

$$y = y_b + \sum_{i=1}^{h} \alpha_i b_i(x), \tag{5}$$

where y_b is a baseline airfoil profile, taken here to be the

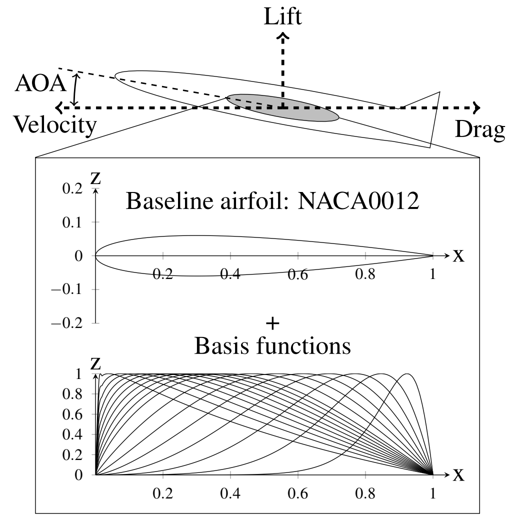

Fig. 6. Schematic layout of the airfoil problem.

Table 6. Percent change of mean results between proposed and reference algorithms.

Function	V1	V2	EAPS	EICMAES
Ack-10	88.8	76.0	580.2	133.1
Gri-10	51.2	59.4	634.6	616.1
Ras-05	46.8	25.7	19.7	234.2
Ros-05	124.6	826.9	1304.2	2405.8
Sch-05	-26.9	-31.8	2.9	-38.7
Wei-10	36.7	24.9	-28.6	-12.4
Ack-20	36.9	203.0	3.3	182.5
Gri-40	22.4	688.5	36.8	3.2
Ras-20	0.2	43.4	82.6	214.4
Ros-20	76.6	40.2	44.5	579.4
Sch-40	16.2	150.4	129.6	115.7
Wei-40	47.3	55.6	7.8	27.4

NACA0012 symmetric airfoil, b_i are geometric basis functions [25] defined as

$$b_i(x) = \left[\sin\left(\pi x^{\frac{\log(0.5)}{\log(i/(h+1))}}\right)\right]^4, \tag{6}$$

and $\alpha_i \in [-0.01, 0.01]$ are the weights to calibrate, namely, the design variables. A low-dimensional case with 6 basis functions and a high-dimensional case with 20 basis functions were examined.

The lift and drag coefficients of candidate airfoils were obtained by using *XFoil* which is a computational fluid dynamics simulation for analysis of subsonic isolated airfoils [26]. Since the objective function required execution of the aerodynamics code each airfoil evaluation required 10-30 seconds while the total duration of an optimization run was in the range of 30 minutes to 1 hour. To ensure structural integrity the thickness (t) between 0.2 to 0.8 of the chord line had to be equal to or larger than a critical value $t^{\star} = 0.1$. Accordingly the objective function used

Table 7. Statistics for the airfoil problem.

		P	V1	V2	EAPS	EICMA
Air06	Mean	**-8.360+1**	-8.048+1	-8.203+1	-7.799+1	-7.231+1
	SD	1.320+1	1.659+1	2.261+1	2.250+0	7.159-1
	Median	**-7.567+1**	-7.533+1	-7.554+1	-7.831+1	-7.264+1
	Min	-1.068+2	-1.268+2	-1.436+2	-8.036+1	-7.290+1
	Max	-7.488+1	-7.174+1	-6.405+1	-7.238+1	-7.099+1
	α					0.01
Air20	Mean	**-3.247+0**	-3.202+0	-3.239+0	-3.174+0	-3.212+0
	SD	6.421-2	6.991-2	8.932-2	8.887-2	9.405-2
	Median	**-3.231+0**	-3.208+0	-3.206+0	-3.142+0	-3.202+0
	Min	-3.354+0	-3.303+0	-3.414+0	-3.348+0	-3.327+0
	Max	-3.151+0	-3.098+0	-3.134+0	-3.070+0	-3.036+0
	α				0.05	

Best mean and median in each test are emphasized.
Air06/20: 6D/20D dimensional airfoil problem, respectively.

Table 8. Mean run-time by number of metamodels.

Function	1 Metamodel	3 Metamodels	% change
Airfoil-06	32.6	30.2	-7.3%
Airfoil-20	35.0	35.2	0.5%

Results are in minutes.

was

$$f = -\frac{c_l}{c_d} + p, \quad p = \begin{cases} \frac{t^\star}{t} \cdot \left|\frac{c_l}{c_d}\right| & \text{if } t < t^\star, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where p is the penalty for violation of the thickness constraint. The flight conditions used were an altitude of 30,000 ft, a speed of Mach 0.75, namely 75% of the speed of sound, while the angle of attack (AOA) was 5° and 15° in the 6D and 20D cases, respectively.

Tests were performed following the setup of Section 4.2 and Table 7 gives the resultant statistics. The performance trends for this problem are consistent with those in Section 4.2 which shows that the proposed algorithm was effective also in this engineering problem.

As in the preceding analysis the impact of the number metamodels being used on the execution time was analyzed and results are given in Table 8. The results are consistent with those observed earlier, namely, there is only a minor variation of the run-time as the number of metamodels increases, an aspect which highlights the efficiency of the proposed algorithm.

In terms of the topologies selected Fig. 7 shows results from a run in the 6D case and in a 20D case, respectively. As in Section 4.2 it follows that no single topology was the overall optimal and that different topologies were selected throughout.

Lastly, Fig. 8 shows the convergence plots from a 6D and from a 20D run, respectively, from which it follows that the proposed algorithm achieved both the fastest convergence and a superior final result.

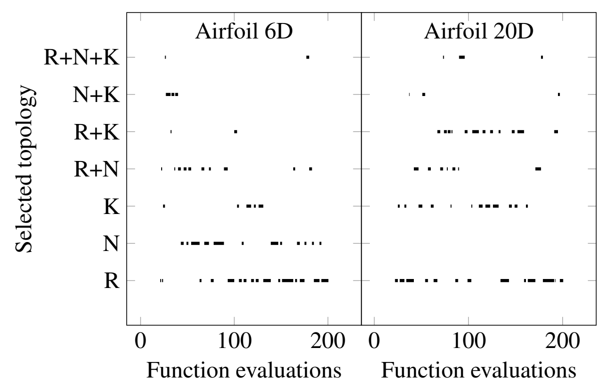

Fig. 7. Selected topologies for the airfoil problems. The abbreviations are: R:RBF, N:RBFN, K:Kriging.

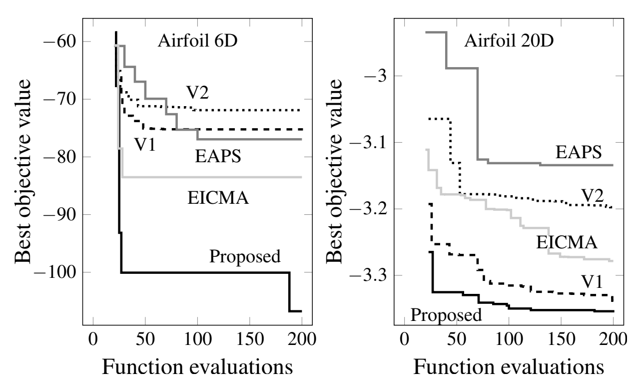

Fig. 8. Convergence plots for two airfoil problems.

5. CONCLUSION

Metamodels are used in computationally expensive black-box optimization problems to address the challenges of a high evaluation cost, a lack of an analytic function expression, and a complicated function landscape. To improve the prediction accuracy ensembles aggregate predictions from several metamodels into a single output. However the optimal ensemble topology, namely, which metamodels it incorporates, is typically unknown a-priori while using a fixed topology may degrade the prediction accuracy.

To address this issue this paper has proposed a new optimization algorithm in which the most suitable topology is selected online during the search. The proposed algorithm also operates within a TR framework to ensure convergence to an optimum of the true expensive black-box function.

In an extensive performance analysis the proposed algorithm was benchmarked against several other candidate algorithms and across a variety of test problems. Analysis shows that: a) the proposed algorithm consistently outperformed the other algorithms in terms of the final result achieved and convergence speed, b) the proposed approach of online selection of the ensemble topology improved the search effectiveness, and c) the optimal ensemble topology varied continuously during the search and no single topology was consistently best. Overall test results show that the proposed algorithm and approach were effective across a range of black-box optimization problems with both a limited number of function evaluations and varied function features and dimensions.

REFERENCES

[1] Y. Tenne and C. K. Goh, eds., *Computational Intelligence in Expensive Optimization Problems*, vol. 2 of *Evolutionary Learning and Optimization*, Springer, Berlin, 2010.

[2] X. Dinh and K. K. Anh, "Radial basis function neural network based adaptive fast nonsingular terminal sliding mode controller for piezo positioning stage," *International Journal of Control, Automation, and Systems*, vol. 15, no. 6, pp. 2892-2905, 2017.

[3] C. Wang, J.-H. Wang, S.-S. Gu, X. Wang, and Y.-X. Zhang, "Elongation prediction of steel-strips in annealing furnace with deep learning via improved incremental extreme learning machine," *International Journal of Control, Automation, and Systems*, vol. 15, no. 3, pp. 1466-1477, 2017.

[4] F. A. C. Viana, R. T. Haftka, and L. T. Watson, "Efficient global optimization algorithm assisted by multiple surrogate technique," *Journal of Global Optimization*, vol. 56, no. 2, pp. 669-689, 2013.

[5] 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.

[6] T. Wortmann, A. Costa, G. Nannicini, and T. Schroepfer, "Advantages of surrogate models for architectural design optimization," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 29, no. 4, pp. 471-481, 2015.

[7] B. Talgorn, S. Le Digabel, and M. Kokkolaras, "Statistical surrogate formulations for simulation-based design optimization," *Journal of Mechanical Design*, vol. 137, no. 2, Paper number MD-14-1128, 2015.

[8] Y. Tenne, "A simulated annealing based optimization algorithm," *Computational Optimization in Engineering - Paradigms and Applications* (H. Peyvandi, ed.), ch. 3, pp. 47-67, IN-TECH Publishing, 2017.

[9] 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.

[10] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, pp. 61-70, 2011.

[11] A. R. Conn and S. Le Digabel, "Use of quadratic models with mesh-adaptive direct search for constrained black-box optimization," *Optimization Methods and Software*, vol. 28, no. 1, pp. 139-158, 2013.

[12] S. Grarton and L. N. Vicente, "A surrogate management framework using rigorous trust-region steps," *Optimization Methods and Software*, vol. 29, no. 1, pp. 10-23, 2014.

[13] I. Couckuyt, F. De Turck, T. Dhaene, and D. Gorissen, "Automatic surrogate model type selection during the optimization of expensive black-box problems," *Proceedings of the 2011 Winter Simulation Conference-WSC 2011* (S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.), (Phoenix, AZ, USA), IEEE, 2011.

[14] D. Gorissen, L. De Tommasi, J. Croon, and T. Dhaene, "Automatic model type selection with heterogeneous evolution: An application to RF circuit block modeling," *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, (Piscataway, New Jersey), pp. 989-996, IEEE, 2008.

[15] Y. Tenne, "An optimization algorithm employing multiple metamodels and optimizers," *International Journal of Automation and Computing*, vol. 10, no. 3, pp. 227-241, 2013.

[16] B. G. M. Husslage, G. Rennen, E. R. van Dam, and D. den Hertog, "Space-filling latin hypercube designs for computer experiments," *Optimization and Engineering*, vol. 12, no. 4, pp. 611-630, 2011.

[17] R. V. Joseph and Y. Hung, "Orthogonal-maximin latin hypercube designs," *Statistica Sinica*, vol. 18, pp. 171-186, 2008.

[18] 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," Technical Report KanGAL 2005005, Nanyang Technological University, Singapore and Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, 2005.

[19] A. Ratle, "Optimal sampling strategies for learning a fitness model," *Proc. of the IEEE Congress on Evolutionary Computation-CEC 1999*, (Piscataway, New Jersey), pp. 2078-2085, IEEE, 1999.

[20] 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.

[21] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Boca Raton, Florida: Chapman and Hall, 4th ed., 2007.

[22] Y.-S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong, "Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems," in *Knowledge Incorporation in Evolutionary Computation* (Y. Jin, ed.) in Studies in Fuzziness and Soft Computing, vol. 167 , pp. 307-332, Springer, Berlin, Heidelberg, 2005.

[23] S. L. Padula, C. R. Gumbert, and W. Li, "Aerospace applications of optimization under uncertainty," *Engineering Optimization*, vol. 7, pp. 317-328, 2006.

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

[25] H.-Y. Wu, S. Yang, F. Liu, and H.-M. Tsai, "Comparison of three geometric representations of airfoils for aerodynamic optimization," *Proc. of the 16th AIAA Computational Fluid Dynamics Conference*, (Reston, Virginia), pp. 1-11, American Institute of Aeronautics and Astronautics, 2003.

[26] M. Drela and H. Youngren, *XFOIL 6.9 User Primer*. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 1 ed., 2001.

Yoel Tenne received his Ph.D. in Mechanical Engineering from Sydney University in 2007. He was an Australian Endeavour postdoctoral fellow at KAIST, Korea in 2009 and a JSPS postdoctoral fellow at Kyoto University, Japan during 2010-2011. His research interests include computational intellligence and applied optimization.