An Algorithm for Expensive Optimization Problems

Yoel Tenne

Department of Mechanical and Mechatronic Engineering
Ariel University, Israel
email: y.tenne@ariel.ac.il

***Abstract*—Computer simulations are used extensively in engineering and science to evaluate candidate designs, as a partial substitute for real-world experiments. *Metamodels*, which are computationally cheaper approximations of the simulation, are often used in these settings to alleviate various issues arising in such simulation-driven design processes. However, due to the high computational cost of running the simulation only a small number of designs can be evaluated, and hence the resultant metamodel will be inaccurate. To achieve a more accurate approximation, *ensembles* employ multiple metamodel variants concurrently, and aggregate their individual predictions into a single one. Nevertheless, the optimal ensemble topology, namely, which types of metamodels should be incorporated, is typically not known a-priori, while using a fixed topology may degrade the search effectiveness. To address this issue, this study proposes a new metamodel-assisted algorithm with *dynamic topology adaptation*, namely, which autonomously adapts the ensemble topology during the search, and dynamically selects the most suitable topology as the search progresses. An extensive performance analysis shows the effectiveness of the proposed algorithm, and highlights the merit of the proposed topology adaptation.**

Keywords–expensive optimization problems; metamodels; ensembles; computational intelligence

I. Introduction

The current availability of high performance computing allows engineers and researchers to evaluate candidate designs with computer simulations instead of using laboratory experiments, thereby reducing the duration and cost of the design process. In this setup, a candidate design is parameterized as a vector of design of variables, and is sent to the simulation for evaluation. Such computer simulations, which still need to be validated with laboratory experiments, transform the design process into an optimization problem having several distinct features [20]:

- The simulation acts as the objective function as it assigns objective values to candidate designs (input vectors), but it is a ‘black-box’, namely, the analytic expression of this mapping is unknown. This can occur since the simulation involves intricate calculations, or the simulation’s code might be inaccessible to the user. In any case, the lack of an analytic expression presents an optimization challenge.
- Each simulation run is often computationally expensive, and hence only a small number of designs can be evaluated.
- Both the real-world physics being modelled, and the numerical simulation process itself, can yield a black-box function with complicated features, such as multiple optima or discontinuities, which add an additional optimization challenge.

An established solution methodology in such scenarios is to incorporate a *metamodel* into the optimization search. The latter is a mathematical approximation of the true expensive function which provides predicted objective values at a much lower computational cost [20]. A variety of metamodels have been proposed, but the optimal type is problem-dependant and is typically not known a-priori. To alleviate this issue, *ensembles* use several metamodels concurrently and aggregate their predictions into a single one [6, 10, 11]. However, the effectiveness of ensembles depends on their topology, namely, which metamodels they incorporate, but again, the optimal topology is typically unknown. To address this issue, this paper proposes an optimization algorithm which dynamically adapts the ensemble topology during the search, such that an optimal ensemble topology is continuously being selected and used. Also, since metamodels are inherently inaccurate, the proposed algorithm operates within a Trust Region (TR) approach to ensure convergence to an optimum of the true expensive function. Performance analysis using both mathematical test functions and a simulation-driven engineering problem shows the effectiveness of the proposed algorithm, and highlights the merit of the proposed dynamic topology adaptation.

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 an extensive performance evaluation. 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 numerical simulation. Metamodels are trained with previously evaluated vectors, and variants include Artificial Neural Networks (ANNs), Kriging, polynomials, and radial basis functions (RBFs), to name a few [11, 17]. A typical metamodel-assisted optimization search begins by sampling an initial set of vectors, followed by a main loop in which a metamodel is trained by using the vectors evaluated so far, seeking an optimum of the metamodel, and evaluating the latter vector, and possibly additional ones, with the true objective function. The process repeats until the number of simulation calls reaches the user-defined limit. Fig. 1 gives a pseudocode of a typical metamodel-assisted algorithm, while more involved frameworks have also been proposed [14, 18].

While metamodels offer several merits, they also introduce new optimization challenges:

- *Prediction inaccuracy*: Since only a small number of vectors can be evaluated with true expensive function the

Figure 1: A typical metamodel-assisted algorithm.

sample an initial set of vectors;
while *stopping criterion not met* **do**
  train a metamodel with the cached vectors;
  seek an optimum of the metamodel;
  evaluate the found solution, and possibly additional vectors, with the true expensive function;
return the best solution found;

resultant metamodel will inherently be inaccurate, and it is therefore necessary to *manage* it to avoid convergence to a poor final result [9]. This can be achieved with the established *Trust Region* (TR) framework [2, 12], in which the search is performed by a series of trial-steps, each confined to the region in which the metamodel is assumed to be sufficiently accurate. The TR is then updated based on the success of the optimization trial step. A strong merit of the TR approach is that it ensures asymptotic convergence to an optimum of the true expensive function [3]. Section III gives a detailed description of the TR approach implemented in this study.

- *Metamodel suitability*: Various metamodel variants have been proposed, but the optimal type is problem-dependant and is typically unknown [7, 18]. Metamodel *ensembles* address this by using multiple metamodels concurrently and aggregating their individual predictions [10, 19]. However, the ensemble topology itself is also problem dependant, and an inadequate topology can degrade the prediction accuracy. As an example, ensembles were generated based on three metamodels: RBFs, radial basis functions neural network (RBFN), and Kriging, as shown in Table I. The respective prediction accuracies of each ensemble was estimated based on the Root Mean Square Error (RMSE) measure across four test functions in dimensions ranging from 5 to 30. It follows that the optimal topology, namely, that having the lowest RMSE, varied across the functions, and that no single topology was the overall best. This suggests that using a fixed ensemble topology is inoptimal, and the following section proposes an algorithm which addresses this issue.

III. Proposed algorithm

The algorithm proposed in this study uses *dynamic topology adaption*, namely, during the search it continuously selects and uses the topology deemed as optimal. The algorithm operates in five main steps, as follows:

Step 1) Initialization: An initial sample of vectors is generated with the Optimal Latin Hypercube Sampling (OLHS) method to obtain a space-filling sample, which in turn improves the prediction accuracy of the metamodels [21].

Step 2) The set of sampled vectors is split into a training and testing set, and the RMSE of each of the $j = 1 \ldots n$ metamodel variants is calculated based on the testing set as follows

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

where $m_j(\boldsymbol{x})$ is a metamodel trained based on the training set, and $\boldsymbol{x}_i, i = 1 \ldots l$ are the testing vectors.

Step 3) The set of sampled vectors is re-split again into training and testing sets, the metamodels are retrained, and the RMSE of each candidate ensemble topology is calculated as follows

$$\varepsilon(\boldsymbol{x}) = \sum_{j=1}^{n} u_j \hat{m}_j(\boldsymbol{x})\,, \quad u_j = \frac{e_j^{-1}}{\sum_{j=1}^{n} e_j^{-1}}\,, \qquad (2a)$$

$$e_\varepsilon = \sqrt{\frac{1}{l}\sum_{i=1}^{l}\big(\varepsilon(\boldsymbol{x}_i) - f(\boldsymbol{x}_i)\big)^2} \qquad (2b)$$

where $\varepsilon(\boldsymbol{x})$ is the ensemble prediction, $\hat{m}_j(\boldsymbol{x})$ is a metamodel trained with the current training set and is active in the topology being evaluated, u_j is the metamodel's weight in the ensemble, while $\boldsymbol{x}_i, i = 1 \ldots l$ are the testing vectors in the current testing set, and e_ε is the RMSE of the ensemble being examined.

Step 4) The ensemble topology with the best (lowest) RMSE is selected for the current iteration. A corresponding ensemble is re-trained based on the selected topology but using all the evaluated vectors.

Step 5) A TR is defined around the current best vector ($\boldsymbol{x}_\text{b}$), and a search is performed to locate the best vector in the TR, based on the ensemble prediction. The search is performed by an evolutionary algorithm (EA) followed by an SQP solver. During this trial search only the ensemble is used, and no calls are made to the expensive function.

Step 6) The best vector found ($\boldsymbol{x}^\star$) is evaluated with the true objective function, and the following updates take place:
- If $f(\boldsymbol{x}^\star) < f(\boldsymbol{x}_\text{b})$: The trial step was successful since the new vector found is indeed better than the current best vector. This implies that the ensemble is accurate, and so the TR is centred at the new vector found and the TR radius is doubled.
- If $f(\boldsymbol{x}^\star) \geqslant f(\boldsymbol{x}_\text{b})$ *and* there are sufficient vectors inside the TR: The trial step failed since the vector found is not better than the current best. This implies that the ensemble is inaccurate, and since there are sufficient vectors in the TR the failure is attributed to the TR being too large. Therefore, the TR radius is halved.
- If $f(\boldsymbol{x}^\star) \geqslant f(\boldsymbol{x}_\text{b})$ *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 the small number of vectors in the TR. Accordingly, a new

TABLE I. The Root Mean Square Error (RMSE) of Different Ensembles Topologies

Function	Ensemble topology			
	R+RN	R+K	RN+K	R+RN+K
Ackley-5D	4.258e-01	3.702e-01	4.151e-01	**2.967e-01**
Rastrigin-10D	1.223e+02	**8.198e+01**	1.312e+02	1.097e+02
Rosenbrock-20D	1.791e+06	1.666e+06	**1.648e+06**	1.693e+06
Schwefel 2.13-30D	**1.882e+06**	2.179e+06	2.343e+06	2.079e+06

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

 ISBN: 978-1-61208-506-7

vector is sampled in a section of the TR which is sparse with vectors.

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. Also, it is important to note that while in this study the metamodels RBF, RBFN, and Kriging were used, the proposed algorithm can accommodate any other type or number metamodels. To complete this section, Fig. 2 presents the pseudocode of the proposed algorithm.

Figure 2: Proposed algorithm with dynamic ensemble adaptation.

```
/* initialization                                              */
generate an initial Optimal Latin Hypercube sample and
evaluate the vectors with the true function;
/* main optimization loop                                      */
repeat
    /* generate a metamodel ensemble                           */
    estimate the prediction error of individual
    metamodels with cross-validation;
    split the vectors evaluated again into a training
    subset and a testing subsets;
    for each candidate ensemble topology do
        calculate the ensemble weights of each
        metamodel in the topology;
        estimate the ensemble accuracy by using the
        testing set;
    select the optimal (most accurate) topology, and
    train an ensemble with all the vectors evaluated;
    /* perform a TR trial step                                 */
    set the TR centre to the best vector found so far;
    perform a trial step (using an EA+SQP) in the TR;
    evaluate the obtained vector with the true expensive
    function;
    /* update the TR                                           */
    if the new solution is better than the current best then
        double the TR radius
    else if the new solution is not better than the current
    best and there the number of vectors in the TR is
    sufficient then
        halve the TR radius;
    else if the new solution is not better than the current
    best and the number of vectors in the TR is
    insufficient then
        add new vectors in the TR to improve the
        prediction accuracy;
until maximum number of simulation calls;
```

IV. Performance analysis

A. Benchmark tests based on mathematical test functions

To assess the effectiveness of the proposed algorithm, it was applied to a well-established set of mathematical test functions [16] which are shown in Table II, in dimensions ranging from 5 to 40.

For a rigorous evaluation, the proposed algorithm was benchmarked against four reference algorithms:

TABLE II. MATHEMATICAL TEST FUNCTIONS

Function	Definition, $f(\boldsymbol{x}) =$	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	$\sum_{i=1}^{d}\{x_i^2/4000\} - \prod_{i=1}^{d}\{\cos\left(x_i/\sqrt{i}\right)\} + 1$	$[-100,100]^d$
Rastrigin	$\sum_{i=1}^{d}\{x_i^2 - 10\cos(2\pi x_i) + 10\}$	$[-5,5]^d$
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	$\sum_{i=1}^{d}\{\sum_{k=0}^{20} 0.5^k\cos(2\pi 3^k(x_i+0.5))\} - d\sum_{k=0}^{20} 0.5^k\cos(\pi 3^k)$	$[-0.5,0.5]^d$

- *V1*: A variant of the proposed algorithm which is identical to it in operation, *except* that it used a single metamodel (RBF), and no ensembles. This algorithm was used to highlight the impact of the ensemble adaptation in comparison to using a fixed metamodel without an ensemble.
- *V2*: A variant of the proposed algorithm which is identical to it in operation, *except* that it used a fixed ensemble which consisted of RBF, RBFN, and Kriging metamodels. This algorithm was used to highlight the impact of the ensemble adaptation in comparison to using a fixed ensemble (no topology adaptation).
- *EA with Periodic Sampling* (EA–PS): A metamodel-assisted algorithm which leverages on the concepts in [4, 13]. 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. This algorithm is representative of many other metamodel-assisted algorithm in the literature.
- *Expected Improvement with Covariance Matrix Adaptation Evolutionary Strategy* (CMA-ES) (EI–CMA-ES): The algorithm combines a CMA-ES optimizer with Kriging metamodels, and uses the expected improvement framework to update the metamodels [1]. This algorithm represents more advanced metamodel-assisted implementations.

These algorithms were chosen as they allowed to evaluate: i) the contribution of dynamic ensemble adaptation (by comparing to the V1 and V2 algorithms), and ii) how the proposed algorithm compares with existing algorithms from the literature. For each algorithm–test function combination 30 runs were repeated so that there were sufficient runs on which a valid statistical analysis could be made. The number of simulations calls, namely, evaluations of the expensive function, was limited to 200, to represent a tight limit on the number of evaluations of the true objective function. Table III gives the resultant test statistics of mean, standard deviation (SD), median, minimum (best) and maximum (worst) objective value in each optimization test case. It also gives the statistic α which indicates the significance level (either 0.05, 0.01) at which the performance of the proposed algorithm was better than that of the other algorithms, where an empty entry indicates that there was no statistically significant performance advantage. The α statistic was determined with the Mann–Whitney nonparametric test [15].

Test results show that the proposed algorithm performed well, as it obtained the best mean statistic in all six cases, and the best median statistic in five out of six cases (all except for the Rastrigin-5 case where it obtained the second best median). Also, its performance had a statistically significant advantage in 13 out of 24 comparisons, namely over 50% of the cases, which further demonstrates its performance advantage. The proposed algorithm also performed well in terms of the SD statistic: it achieved the best (lowest) SD in 3 cases, and was comparable to the best performing algorithms in other cases, which shows that it typically maintained a low level of variability in its performance, which is also desirable.

The test results also highlight the merits of the dynamic topology adaptation approach of adapting the ensemble topology, as evident from the performance gains with respect to using a single metamodel (V1 algorithm) or a fixed ensemble (V2 algorithm). The proposed algorithm also outperformed the two reference algorithms from the literature, which shows that it was competitive with existing approaches.

The analysis also examined the pattern of updates of the ensemble topology to study if one specific topology was mainly selected, or if various topologies were used. Accordingly, Fig. 3 shows plots of the dynamic ensemble adaptation from a run with the Ackley-10D function and another with the Rosenbrock-20D function. While a Kriging metamodel topology was selected more frequently than the other topologies, in both tests the optimal topology varied consistently throughout the search. This further highlights the merit of the proposed topology adaptation approach over that of using a fixed topology.

B. Engineering test problem

Beyond the tests with mathematical test functions, the numerical experiments also included a test based on a simulation-driven engineering problem, to more closely represent real-world problems. The optimization goal here was to find an airfoil shape which maximizes the lift produced while minimizing the drag (aerodynamic friction) at some prescribed flight conditions. Candidate airfoils were represented with the method of Hicks and Henne [8], such that an airfoil profile was given by

$$y = y_b + \sum_{i=1}^{h} \alpha_i b_i(x)\,, \tag{3a}$$

$$b_i(x) = \left[\sin\left(\pi x^{\frac{\log(0.5)}{\log(i/(h+1))}}\right)\right]^4\,, \tag{3b}$$

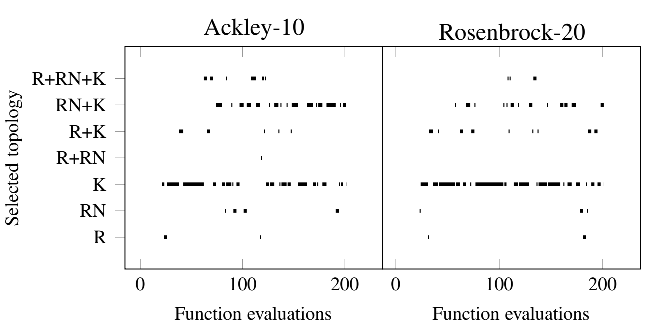

Figure 3. Selected ensemble topologies (R:RBF, RN:RBFN, K:Kriging).

where y_b is a baseline airfoil profile, taken here to be the NACA0012 symmetric airfoil, b_i are geometric basis functions [22], and $\alpha_i \in [-0.01, 0.01]$ are weights whose optimal values need to be found, namely, those which define the best performing airfoil. To visualize the problem formulation, Fig. 4 shows the layout of the airfoil problem.

Two optimization scenarios were examined: 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 [5]. To ensure structural integrity of the airfoil, the minimum airfoil thickness (t) between 20% to 80% of the airfoil chord line needed to be no less than a critical value $t^\star = 0.1$. Accordingly, the objective function used was

$$f = -\frac{c_l}{c_d} + p\,, \quad p = \begin{cases} \frac{t^\star}{t} \cdot \left|\frac{c_l}{c_d}\right| & \text{if } t < t^\star \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where p is a penalty for violation of the thickness constraint. The prescribed flight conditions were a cruise altitude of 30,000 ft, a cruise speed of Mach 0.7, namely 70% of the speed of sound, and an angle of attack (AOA) of 2° , which is the angle between the airfoil chord line and the aircraft velocity.

Tests were performed following the setup in Section IV-A, and Table IV gives the resultant test statistics. It follows that the results obtained here are consistent with those of the previous section, and that the proposed algorithm again outperformed the others algorithms, as evident from the test statistics.

Also following Section IV-A, Fig. 5 shows the ensemble topologies which were selected during one run from the 6D scenario and one from the 20D scenario, respectively. As before, the optimal topology varied continuously during the

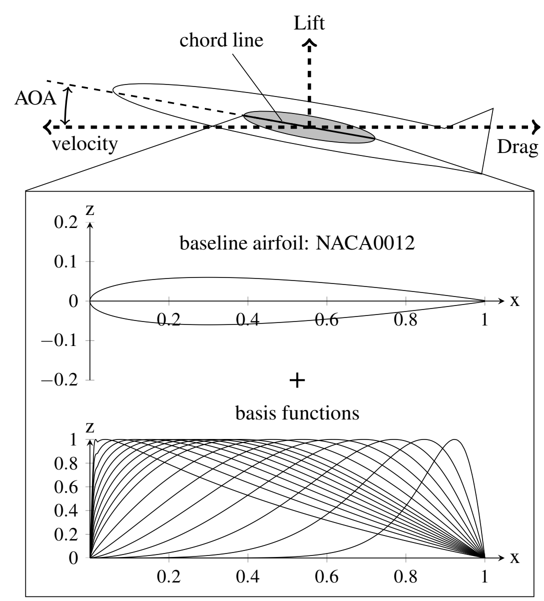

Figure 4. The layout of the airfoil optimization problem.

TABLE III. TEST STATISTICS–MATHEMATICAL TEST FUNCTIONS

		Proposed	V1	V2	EA–PS	EI–CMA-ES
Ackley-10	Mean	**7.705e+00**	1.455e+01	1.356e+01	5.241e+00	1.796e+01
	SD	8.359e+00	4.649e+00	8.051e+00	5.590e-01	1.529e+00
	Median	**2.314e+00**	1.592e+01	1.908e+01	5.408e+00	1.797e+01
	Min(best)	9.007e-02	2.383e+00	3.457e+00	4.098e+00	1.443e+01
	Max(worst)	1.836e+01	1.825e+01	2.048e+01	6.010e+00	1.988e+01
	α			0.01		0.01
Griewank-10	Mean	**1.304e-01**	1.972e-01	2.078e-01	9.579e-01	9.338e-01
	SD	1.851e-01	1.714e-01	2.213e-01	1.076e-01	2.435e-01
	Median	**7.747e-02**	1.294e-01	1.357e-01	9.862e-01	1.007e+00
	Min(best)	9.350e-03	3.569e-02	2.290e-02	7.146e-01	2.441e-01
	Max(worst)	6.505e-01	5.661e-01	7.601e-01	1.046e+00	1.050e+00
	α				0.01	0.01
Rastrigin-5	Mean	**6.377e+00**	9.360e+00	8.018e+00	7.631e+00	2.131e+01
	SD	3.728e+00	7.852e+00	8.349e+00	4.811e+00	4.890e+00
	Median	5.980e+00	7.464e+00	**4.298e+00**	7.226e+00	2.139e+01
	Min(best)	1.997e+00	1.005e+00	3.369e+00	1.621e+00	1.353e+01
	Max(worst)	1.195e+01	2.787e+01	3.076e+01	1.456e+01	3.006e+01
	α					0.01
Rosenbrock-20	Mean	**5.839e+02**	1.031e+03	8.186e+02	8.435e+02	3.967e+03
	SD	2.094e+02	5.818e+02	3.823e+02	3.012e+02	9.406e+02
	Median	**5.956e+02**	8.665e+02	7.932e+02	7.782e+02	3.685e+03
	Min(best)	2.143e+02	5.483e+02	3.078e+02	4.676e+02	3.141e+03
	Max(worst)	8.905e+02	2.517e+03	1.521e+03	1.439e+03	6.144e+03
	α		0.01		0.05	0.01
Schwefel-40	Mean	**7.727e+05**	8.981e+05	1.935e+06	1.774e+06	1.667e+06
	SD	2.219e+05	2.571e+05	6.789e+05	2.509e+05	6.520e+05
	Median	**7.243e+05**	8.622e+05	2.032e+06	1.744e+06	1.528e+06
	Min(best)	5.130e+05	5.885e+05	8.715e+05	1.415e+06	8.933e+05
	Max(worst)	1.131e+06	1.362e+06	3.065e+06	2.104e+06	2.871e+06
	α			0.01	0.01	0.01
Weierstrass-40	Mean	**2.824e+01**	4.160e+01	4.394e+01	3.045e+01	3.598e+01
	SD	4.401e+00	4.261e+00	3.885e+00	1.645e+00	1.463e+01
	Median	**2.547e+01**	4.227e+01	4.461e+01	2.995e+01	2.597e+01
	Min(best)	2.421e+01	3.353e+01	3.726e+01	2.878e+01	2.100e+01
	Max(worst)	3.482e+01	4.794e+01	4.867e+01	3.337e+01	5.817e+01
	α		0.01	0.01		

entire search. These results, combined with the test statistics, show that dynamically adapting the ensemble topology during the search improved the search effectiveness also in these simulation-driven problems.

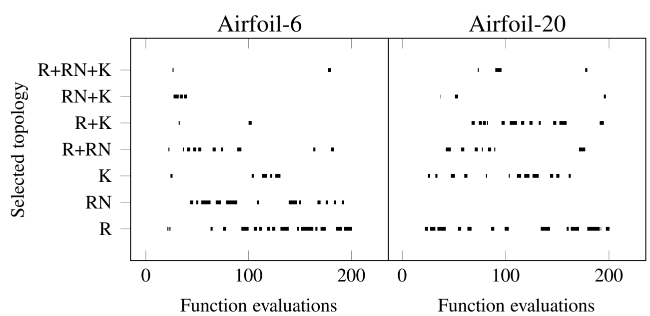

Figure 5. Selected topologies for the airfoil problems (R:RBF, RN:RBFN, K:Kriging).

V. CONCLUSION AND FUTURE WORK

The use of simulations in engineering design defines a unique optimization problem which is termed in the literature as an *expensive black-box* problem. Metamodels are used in such settings to approximate the computationally expensive simulation, and to allow a more efficient optimization search. Since the optimal metamodel variant is problem-dependant and is typically unknown a-priori, ensembles use multiple metamodels concurrently, and aggregate their predictions to a single output, in an attempt to improve the prediction accuracy. However, the ensemble topology itself is also problem-dependant, and the optimal topology is also typically unknown. To address this issue, this study has proposed an ensemble based optimization algorithm which dynamically adapts the ensemble topology during the search, so that an optimal topology is used at each stage. Furthermore, the proposed algorithm operates within a TR framework to ensure convergence to an optimum of the true expensive function in spite of the inherent metamodel prediction inaccuracies. In a detailed performance analysis the proposed algorithm was benchmarked against various algorithms with no dynamic topology adaptation. It consistently outperformed the other algorithms across the different test problems, and the optimal topology varied continuously throughout the search. Overall, results show that the proposed algorithm performed well across a range of test problems, and that the effectiveness of metamodel-assisted search was improved with the proposed dynamic topology adaption. Based on the promising results obtained, future work will examine additional topology selection mechanisms, for example, such as those based on other error measures or other sampling approaches.

TABLE IV. TEST STATISTICS–AIRFOIL PROBLEM

		Proposed	V1	V2	EA–PS	EI–CMA-ES
6D	Mean	**-8.360e+01**	-8.048e+01	-8.203e+01	-7.799e+01	-7.231e+01
	SD	1.320e+01	1.659e+01	2.261e+01	2.250e+00	7.159e-01
	Median	**-7.567e+01**	-7.533e+01	-7.554e+01	-7.831e+01	-7.264e+01
	Min(best)	-1.068e+02	-1.268e+02	-1.436e+02	-8.036e+01	-7.290e+01
	Max(worst)	-7.488e+01	-7.174e+01	-6.405e+01	-7.238e+01	-7.099e+01
	α					0.01
20D	Mean	**-3.247e+00**	-3.202e+00	-3.239e+00	-3.174e+00	-3.212e+00
	SD	6.421e-02	6.991e-02	8.932e-02	8.887e-02	9.405e-02
	Median	**-3.231e+00**	-3.208e+00	-3.206e+00	-3.142e+00	-3.202e+00
	Min(best)	-3.354e+00	-3.303e+00	-3.414e+00	-3.348e+00	-3.327e+00
	Max(worst)	-3.151e+00	-3.098e+00	-3.134e+00	-3.070e+00	-3.036e+00
	α				0.05	

REFERENCES

[1] 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.

[2] 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.

[3] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*. Philadelphia, Pennsylvania: SIAM, 2000.

[4] K. A. de Jong, *Evolutionary Computation:A Unified Approach*. Cambridge, Massachusetts: MIT Press, 2006.

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

[6] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo, "Ensembles of surrogates," *Structural and Multidisciplinary Optimization*, vol. 33, pp. 199–216, 2007.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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, pp. 123–144, 2014.

[12] M. J. D. Powell, "On trust region methods for unconstrained minimization without derivatives," *Mathematical Programming, Series B*, vol. 97, pp. 605–623, 2003.

[13] A. Ratle, "Optimal sampling strategies for learning a fitness model," in *The 1999 IEEE Congress on Evolutionary Computation–CEC 1999*. Piscataway, New Jersey: IEEE, 1999, pp. 2078–2085.

[14] 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, pp. 1719–1753, 2013.

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

[16] 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.

[17] Y. Tenne, "A computational intelligence algorithm for simulation-driven optimization problems," *Advances in Engineering Software*, vol. 47, pp. 62–71, 2012.

[18] ——, "An optimization algorithm employing multiple metamodels and optimizers," *International Journal of Automation and Computing*, vol. 10, no. 3, pp. 227–241, 2013.

[19] ——, "An algorithm for computationally expensive engineering optimization problems," *International Journal of General Systems*, vol. 42, no. 5, pp. 458–488, 2013.

[20] Y. Tenne and C. K. Goh, Eds., *Computational Intelligence in Expensive Optimization Problems*, ser. Evolutionary Learning and Optimization. Berlin: Springer, 2010, vol. 2.

[21] 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.

[22] 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 16th AIAA Computational Fluid Dynamics Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2003, AIAA 2003-4095.