Proc. of the Third Intl. Conf. on Advances in Mechanical and Automation Engineering - MAE 2015.

ISBN: 978-1-63248-080-4 doi: 10.15224/ 978-1-63248-080-4-111

# A framework for engineering design optimization

[ Yoel TENNE ]

***Abstract*—In modern engineering practice computer simulations are often used as a substitute for laboratory experiments. This setup yields a simulation-driven optimization problem in which the computer simulation acts as the objective function. Since simulation runs are typically computationally expensive, metamodels are used to approximate the simulation. However, this setup can face difficulties in high-dimensional problems as the accuracy of metamodels then becomes very poor. To address this issue this paper proposes an optimization framework which incorporates a dimensionality-reduction technique into the search. This allows to formulate a valid lower dimensional problem which is easier to solve. The solution found is then mapped back to the original high dimensional space. Performance analysis with an airfoil shape optimization problem shows the effectiveness of the proposed framework. *(Abstract)***

***Keywords—engineering design optimization, computer simulations, metamodels, high-dimensionality (key words)***

## I. Introduction

In modern engineering computer simulations are often used as a substitute for laboratory experiments when evaluating candidate designs. This setup enhances the design process in several aspects, for example by reducing its duration or cost. Such simulations, which need to be validated with laboratory experiments, act as the problem's objective function since they assign a merit value to each candidate design. This in turn formulates an optimization problem where candidate designs are represented as vectors of design variables, and the goal is then to find the variables values which yield the best performing design [1,18,20]. It follows that under these settings a candidate design and a vector of design variables are equivalent, and are used interchangeably in this paper. While computer simulations have numerous merits, they also introduce several challenges into the design process: i) each simulation run is *computationally-expensive*, namely, it requires large computational resources and hence only a small number of simulation runs can be made, ii) there is no analytic expression which defines how candidate designs are mapped to their merit value, and this effectively renders the simulation as a *black-box function*, and iii) the latter black-box function can have a complicated landscape with multiple local optima, an aspect which further exacerbates the optimization difficulty.

In such problem variants classical optimization approaches will often struggle and this has motivated the development of new techniques tailored for such simulation-driven problems [1,2,18]. One such established framework is that of using *metamodels*, also termed in the literature as *surrogates* or *response surfaces*, which are used as a computationally cheaper approximation of the true expensive function (namely, the simulation). This way, approximate objective values can be obtained with far fewer computer resources and therefore more designs can be evaluated.

While the metamodel-assisted framework has proven to be effective, real-world optimization problems are often characterized by a large number of design variables, namely, they are high-dimensional. In such settings the prediction accuracy of the metamodels becomes poor, which in turn degrades the effectiveness of the optimization process [1,9].

To address this issue this paper proposes an optimization framework which incorporates a *dimensionality-reduction* technique into the search, such that during the optimization process the framework formulates a valid lower-dimensional problem. The latter involves fewer design variables and therefore the prediction accuracy of the metamodels improves. The solution obtained from these simplified problems is then mapped back into the original high-dimensional space to obtain a new optimal design. The effectiveness of the proposed approach is demonstrated by numerical experiments with an engineering problem of airfoil shape optimization. The remainder of this paper is organized as follows: Section 2 provides the general background and literature survey, Section 3 describes the proposed framework, and Section 4 presents the numerical experiments and their analysis. Lastly, section 5 concludes this paper.

## II. Background

As mentioned in Section 1, simulation-driven optimization problems are characterized by function evaluations which require large computational resources. This in turn severely restricts the number of candidate designs which can be evaluated and hampers the effectiveness of the optimization process. An established approach to circumvent this limitation is by using *metamodels*, also termed in the literature as *surrogate models* or *response surfaces* [1,18,20]. These are approximations of the true expensive simulation, but which are computationally cheaper to evaluate. During the optimization process estimated objective values are obtained from the metamodel at a significantly lower computational cost, which allows to evaluate a larger number of candidate designs. Metamodels are typically interpolants and are trained based on previously evaluated vectors. Some common variants include quadratic or higher order polynomials, radial basis functions (RBF), Kriging, and artificial neural networks (ANN) [1,18,20].

After training a metamodel an optimizer is invoked to locate a good solution based on the metamodel predictions. Since an effective search needs to be both global and local, a common approach is to use both a global optimizer which explores the metamodel landscape globally, combined with a local optimizer which can refine solutions and yield more accurate results. Common choices for the global optimizers include evolutionary algorithms (EA), particle swarm optimizers (PSO), and simmulated annealing (SA), while the some local optimizer include the Nelder-Mean simplex, sequential quadratic programming (SQP), and the hill climbing algorithm [1,2,18,20].

While such metamodel-assisted framework perform well in problems with a small number of variables, they can struggle when the number of variables increases (over ten) since then the prediction accuracy of the metamodel becomes poor. To illustrate this, a Kriging metamodel [14,20] was trained to approximate the Rastrigin function in dimensions 2, 20, 50, and 100. Afterwards a random sample of 500 vectors was generated and the metamodel was evaluated at these vectors. The obtained responses were sorted to produce histograms, which are shown in Figure 1.

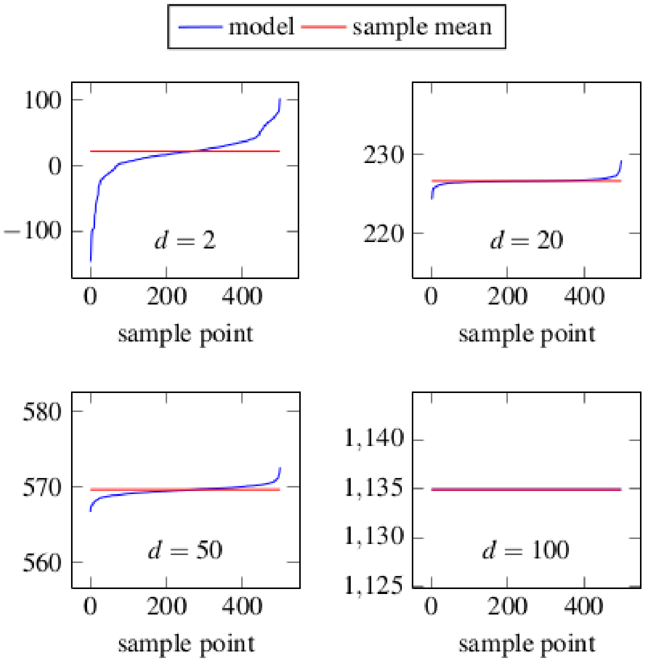

Figure 1: Historgrams of the responses of Kriging metamodels in dimensions 2, 20, 50, and 100.

It follows that as the function dimension increased the metamodel lost its ability to capture the actual function features and was basically reduced to a flat plane at the mean level of the sample response. A similar effect was observed with other common metamodels such as RBF and the Shepard interpolant, which also use an interpolation mechanism based on the distance from the sample vectors. Overall, it follows that in high-dimensional problems the metamodel approximation became poor which in turn would hamper any optimization process. To combat this issue the following section describes an optimization algorithm which incorporates dimensionality-reduction into the search.

# III. Proposed framework

## A. Dimensionality-reduction

The goal of the dimensionality-reduction (DR) stage is to formulate an optimization problem which is simpler to solve, but whose solution can be mapped back to a good solution of the original high-dimensional problem. The DR approach implemented in this study was that of *variable selection* (VS), also known in the literature as *subset/feature selection* [12,13]. The goal of VS is to identify a subset of the original problem variables which are the most important. The approach differs from other DR methods such as principal component analysis (PCA) since the reduced subset consists of some of the original problem variables while in methods such PCA the variables in the reduced subset are linear combinations of the original variables, and may not be easily associated with the problem formulation. VS has been successfully used in diverse domains such as chemistry, ergonomics and machine learning [9,10,12,13,16,17].

In this study VS is implemented by means of *cross-validation*, namely, the set of sampled vectors is split into a *training set* and a *testing set.* Assuming the original problem dimension is $d$and the reduced dimension is $\bar{d}$, then the lower dimensional vector is obtained by discarding $d-\bar{d}$elements in each vector. Since there are different variable combination it is necessary to explore several subsets. The projected vectors are split into a training set and a testing set. A metamodel is trained by using the former and is then tested on the latter. The prediction accuracy of the resultant metamodel is gauged by the mean squared error

$$MSE = \frac{1}{n}\sum|m(x_i) - f(x_i)|^2$$

where $n$ is the number of vectors in the testing sample, $m(x)$ is the response of the metamodel and $f(x)$is the true function value, where $x_i$is a vector from the testing set. The process is repeated for different subsets, namely, different sets of variables, and the subset which yields the minimal MSE is chosen is the active set.

## B. Framework workflow

The proposed framework follows the global-local layout outlined in Section 2, but incorporates dimensionality-reduction by introducing two new elements:

a) A *working dimension* $\bar{d} < d$, which is the dimension prescribed by the user for the lower-dimensional problem, and

b) A *subset vector* $S \in R^d$which indicates which variables of the original problem which has dimension $d$are active, namely, have been selected, to participate in the lower dimensional problem. Specifically, $S_i = 1$ indicates that the *i*th variable is active, while $S_i = 0$indicates that it is not.

The framework begins by generating an initial sample of vectors by using a Latin hypercube design [15,19], to obtain a space-filling sample as this improves the prediction accuracy of the metamodel. The sampled vectors are then evaluated with the true expensive function and the vectors are cached.

Next the main optimization loop begins where the sampled vectors are projected to the working dimension $\bar{d}$, and this yields a lower dimensional sample. A metamodel is then trained based on the latter sample, and in this study the well-established Kriging metamodel was used [14]. Metamodel predictions are inherently inaccurate and this can hamper the optimization search and lead to convergence to a poor solution. To address this the proposed framework implements a trust-region (TR) approach [2,7], namely, it searches for an optimum of the metamodel in a confined region (the trust-region) which is centred at the current best solution. An optimum is search for by using an evolutionary algorithms (EA) which explores the metamodel landscape [3], and is then followed by a gradient-based SQP solver to refine the obtained solution. The low-dimensional solution found is then mapped to the original high-dimensional solution where the variables which were inactive are assigned the corresponding values from the best solution found so far, namely

$$x_i^H = \begin{cases} x_i^L & if x_i isactive \\ x_i & if x_i isinactive \end{cases}$$

where $x^H$ is the solution mapped back to the high dimension, $x^L$is the obtained low-dimensional solution, and $x^*$is current best high-dimensional solution found. Based on the obtained value of the evaluated solution the trust-region is updated as follows (assuming a minimization problem):

- if $f(x^H) < f(x^*)$: The optimization step succeeded as the new solution found is better than best found so far. Since the metamodel appears to be accurate the TR radius is doubled and is centred around the new best solution.
- if $f(x^H) \geq f(x^*)$and the number of vectors in the trust-region is deemed as too small: the optimization step failed, but this may be due to the metamodel being inaccurate because of too few sample vectors in the TR. Therefore another vector is sampled in the TR to improve the prediction accuracy.
- if $f(x^H) \geq f(x^*)$and the number of vectors in the TR is deemed as sufficient: the optimization step failed, but because there the are sufficient vectors in the trust region the failure is attributed to the TR being too large. Therefore, the trust-region radius is halved.

After these steps the entire process repeats, until the prescribed limit of simulation runs is reached. To conclude the framework description Algorithm 1 gives its workflow. It is emphasized that while in this study the proposed framework used a Kriging metamodel, any other metamodel variant can be readily used.

```
generate an initial sample of vectors and evaluate them with the true objective function;
define a trust region around the best vector;
repeat
    select the variables for the lower dimensional space;
    project the sampled vectors to the lower dimension;
    train a metamodel with the projected sample;
    perform a trust-region optimization;
    project the obtained solution to the high-dimension;
    update the trust-region and sample;
until the limit of simulation runs is reached;
```

Algorithm 1: The workflow of the proposed framework

# IV.Performance analysis

To evaluate its effectiveness, the proposed framework was applied to an engineering problem of airfoil shape optimization. The goal was to simultaneously optimize the shape of five airfoils at different locations along an aircraft wing. Each airfoil was subject to a different geometric twist (rotation) thereby increasing its effective angle of attack (AOA), which is the angle between the airfoil chord and the velocity vector. The twist angles per airfoil were $1^o, 2^o, 3^o, 4^o, and\ 5^o$, respectively. The goal was to maximize the lift coefficient $(c_L)$ while minimzing the aerodynamic drag coefficient $(c_D)$. Airfoils were represented with the Hicks-Henne parametrization [4] which uses a baseline airofil shape $y_b$ (taken here to be the NACA0012 symmetric airfoil) and adds shape functions

$$b_i(x) = \left[sin\left(\pi x^{\frac{log0.5}{log(i/(h+1))}}\right)\right], i = 1\ldots h$$

such that the resultant airfoil shape is given by

$$y = y_b + \sum \alpha_i\, b_i(x)$$

where $\alpha_i \in [-1,1]$are coefficients to determine, namely, the design variables and $h$is the prescribed number of basis functions. In this study $h = 25$was used for each upper/lower airfoil curve was 25 which resulted in 50 functions per airfoil and a total of 250 variables (50 × 5 airfoils). The flight conditions were set to a cruise altitude of 30,000 feet, a cruise speed of 70% of the speed of sound and an angle-of-attack (AOA) of $2^o$.

The objective function used was the mean performance over the five airfoils

$$f = \frac{1}{5}\sum\left(-\frac{c_L}{c_D} + p\right),$$

where $p$ is a penalty for violating the thickness constraint and was defined as

$$p = \begin{cases} \left|\frac{cL}{CD}\right| \frac{t*}{t} & \text{if t<t*} \\ 0 & \text{otherwise} \end{cases}$$

where $t$ is the airfoil thickness between 0.2 to 0.8 of the chord line and $t^* = 0.1$ is the minimum allowed thickness. Airfoils were evaluated with Xfoil aerodynamic analysis code for subsonic airfoils [5]. Figure 2 gives the layout of the airfoil problem.

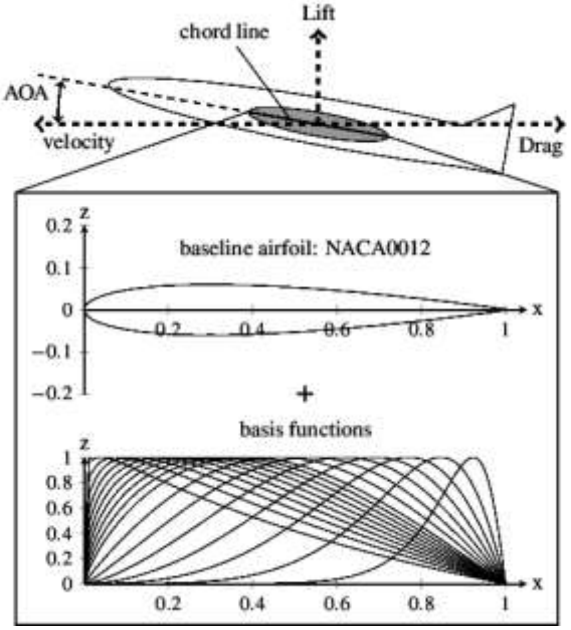

Figure 2: Formulation of the airfoil problem and the Hicks-Henne parametrization for a single airfoil.

For benchmarking purposes the proposed framework was compared to the EA-PS algorithm [6] which combines an evolutionary algorithm with a Kriging metamodel, and uses periodic sampling to update the metamodel and ensure its accuracy. Table 1 gives the resultant test statistics from which it follows that the proposed framework had a statistically significant performance advantage at the 0.01 significance level. To visualize results Figure 3 compares the airfoils obtained in one representative test case (blue/dark gray) and overlaid with the baseline NACA0012 airfoils (yellow/light gray), the rotation angles have been enlarge for emphasis. There is a thickening of the upper curve and tapering in the lower curve which both contribute to an increased lift.

Table 1: Test statistics

| Statistic | Proposed framework | Reference algorithm EA-PS |
|---|---|---|
| Mean | -6.599e+02 | -8.073e+01 |
| Median | -9.33e+01 | -8.109e+01 |
| α – significance level | | 0.01 |

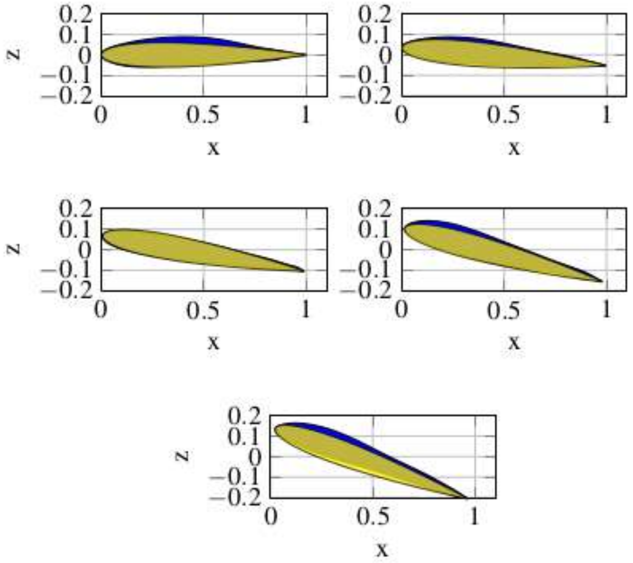

Figure 3: Resultant airfoils.

# V. Conclusion

Real-world engineering design optimization problems are often solved by using computer simulations. Metamodel-assisted frameworks which are commonly used in such settings can struggle when the number of design variables is high, namely the problem becomes high dimensional.

To address this issue this study has proposed a framework which incorporates dimensionality-reduction into the optimization loop. This is achieved by incorporating a variable selection technique into the optimization search, such that the proposed framework identifies at each iteration a subset of the original variables which are deemed as the most important. The optimization process is then performed in a lower dimensional space and solutions are projected back into the original high dimensional space for evaluation. Performance analysis with a simulation-driven engineering problems shows the effectiveness of the proposed framework.

ISBN: 978-1-63248-080-4 doi: 10.15224/ 978-1-63248-080-4-111

## References

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

[2] Y. Tenne and S. W. Armfield, "A framework for memetic optimization using variable global and local surrogate models," Journal of Soft Computing, vol. 13, no. 8, pp. 781-793, 2009.

[3] A. Chipperfield, P. Fleming, H. Pohlheim, and C. Fonseca, Genetic Algorithm TOOLBOX For Use with MATLAB, Version 1.2. Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, 1994.

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

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

[6] A. Ratle, "Optimal sampling strategies for learning a fitness model," in The 1999 IEEE Congress on Evolutionary Computation-CEC 1999, (Piscataway, New Jersey), pp. 2078-2085, IEEE, 1999.

[7] A. R. Conn, K. Scheinberg, and L. N. Vicente, Introduction to Derivative-Free Optimization (Mps-Siam Series on Optimization. Philadelphia, Pennsylvania: SIAM, 2009.

[8] A. Herrero and M. Cruz Ortiz, "Qualitative and quantitative aspects of the application of genetic algorithm-based variable selection in polarogrpahy and stripping voltammetry," Analytica Chemica Acta, vol. 378, pp. 245-259, 1998.

[9] S. H. Han and H. Yang, "Screening important design variables for building a usability model: Genetic algorithm-based partial least squares approach", International Journal of Industrial Ergonomics, vol. 33, pp. 159-171, 2004.

[10] M. Cui, S. Prasad, M. Mahrooghy, L. M. Bruce, and J. Aanstoos, "Genetic algorithms and linear discriminant analysis based dimensionality reduction for remotely sensed image analysis," in Geoscience and Remote Sensing Symposium(IGARSS), 2011, pp. 2373-2376, IEEE, 2011.

[11] J. Yin, Y. Wang, and J. Hu, "A new dimensionality reduction algorithm for hyperspectral image using evolutionary strategy," IEEE Transactions on Industrial Informatics, vol. 8, no. 4, pp. 935-943, 2012.

[12] W. M. Hartmann, "Dimension reduction vs. variable selection," in PARA 2004,vol. 3732 of Lecture Notes in Computer Science, pp. 931-938, Springer, 2004.

[13] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," The Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[14] L. N. Soren, H. B. Nielsen, and J. Sondergaard, "Aspects of the MATLAB toolbox DACE," Technical Report IMM-TR-2002-13, Informatik and Mathematical Modelling, Technical University of Denmark, Lingby,Copenhagen, 2002.

[15] 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.

[16] T. Santhanam and M. S. Padmavathi, "Application of K-Means and Genetic Algorithms for Dimension Reduction by Integrating SVM for Diabetes Diagnosis," Procedia Computer Science, vol. 47, pp. 76-83, 2015.

[17] R. S. Babatunde, S. Olabiyisi, and E. Omidiora, "Feature dimensionality reduction using a dual level metaheuristic algorithm," International Journal of Applied Information Systems, vol. 7, no. 1, pp. 49-52, 2014.

[18] Y. Tenne, K. Izui, and S. Nishiwaki, "An adaptive computational intelligence algorithm for simulation-driven optimization problems," International Journal on Advances in Software, vol. 5, no. 1 and 2, pp. 131-145, 2012. invited submission.

[19] Y. Tenne, "Initial sampling methods in metamodel-assisted optimization,"Engineering with Computers, vol. 31, no. 4, pp. 661-680, 2014.

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