AIAA JOURNAL
Vol. 46, No. 5, May 2008

Kriging Hyperparameter Tuning Strategies

David J. J. Toal,* Neil W. Bressloff,† and Andy J. Keane‡
University of Southampton, Southampton, SO17 1BJ England, United Kingdom

DOI: 10.2514/1.34822

Response surfaces have been extensively used as a method of building effective surrogate models of high-fidelity computational simulations. Of the numerous types of response surface models, kriging is perhaps one of the most effective, due to its ability to model complicated responses through interpolation or regression of known data while providing an estimate of the error in its prediction. There is, however, little information indicating the extent to which the hyperparameters of a kriging model need to be tuned for the resulting surrogate model to be effective. The following paper addresses this issue by investigating how often and how well it is necessary to tune the hyperparameters of a kriging model as it is updated during an optimization process. To this end, an optimization benchmarking procedure is introduced and used to assess the performance of five different tuning strategies over a range of problem sizes. The results of this benchmark demonstrate the performance gains that can be associated with reducing the complexity of the hyperparameter tuning process for complicated design problems. The strategy of tuning hyperparameters only once after the initial design of experiments is shown to perform poorly.

Nomenclature

A	=	amplitude of the Hicks–Henne function
C_p	=	pressure coefficient
c	=	chord length, m
d	=	number of variables
I	=	identity matrix
i	=	sample point index
j	=	sample point index
ℓ	=	variable index
n	=	number of sample points
p	=	hyperparameter determining smoothness
$\boldsymbol{R}$	=	correlation matrix
r	=	linear correlation coefficient
t	=	Hicks–Henne function sharpness
x	=	variable
x_p	=	maximum of the Hicks–Henne function
Y	=	random variables
y	=	objective function value
β	=	mean
θ	=	hyperparameter determining correlation
λ	=	regression constant
σ	=	standard deviation

I. Introduction

EVEN in the modern world of high-performance computing, it is rarely feasible to exhaustively search a design space using high-fidelity computer simulations. The use of surrogate models (otherwise known as response surfaces or metamodels) within the design optimization process therefore remains an important approach. Of the many different surrogate models available, such as simple polynomials or radial basis functions, the popularity of kriging has grown due to its ability to effectively represent complicated responses while providing an error estimate of the predictor.

Received 28 September 2007; revision received 7 January 2008; accepted for publication 7 January 2008.

*Graduate Research Student, School of Engineering Sciences. Student Member AIAA.

†Senior Research Fellow, School of Engineering Sciences.

‡Professor of Computational Engineering, School of Engineering Sciences.

Kriging was first used by geologists to estimate mineral concentrations within a particular region, but has since been used in the creation of surrogate models of deterministic computational experiments, a process popularized by Sacks et al. [1]. Various aerodynamic [2,3], structural [3–5], and multiobjective [6,7] problems have all been investigated using kriging, and the basic kriging theory has been elaborated to include regression [8], as well as secondary data, through cokriging [9].

Although the performance of kriging has been extensively compared with that of other surrogate models [10–13], and a number of different tuning techniques have been compared [14], there is little information in the literature on the effect that the degree of hyperparameter tuning has on an optimization process. Throughout the literature, it is consistently observed that the time spent tuning the hyperparameters of a kriging model can be quite significant and, for high-dimensional problems with many data points, comparable with the cost of the high-fidelity simulations used to obtain the objective function values. Hyperparameter tuning can be especially costly if an updating process is used to improve the surrogate model, as in [7]. Tuning of the hyperparameters effectively becomes a bottleneck in the optimization process, introducing a significant delay before new updates can be evaluated.

The current paper attempts to illuminate the issue of hyperparameter tuning by assessing the performance of five different tuning strategies when applied to the inverse design of the supercritical RAE-2822 airfoil. The geometry parameterization employed in this design problem allows a systematic increase in the number of variables used in the optimization process while maintaining continuity between the geometries as the number of variables is changed.

II. Benchmarking Procedure Overview

The presented benchmarking procedure employs the Options Design Exploration System [15] and, at its core, consists of the basic response surface construction and updating process popular throughout the literature [2,5,7,8]. This process begins with a sampling of the optimization problem, from which a surrogate model is constructed in an attempt to accurately reproduce the response of the objective function to changes in the variables. In the case of kriging, the construction of the surrogate model requires the selection of an optimal set of hyperparameters to insure that the model accurately represents the design space. It is this process of the selection of an optimal set of hyperparameters that is henceforth referred to as tuning. With a surrogate model constructed, it can be searched to find regions of the design space that will produce optimal designs. Typically, a global optimizer such as a genetic algorithm

(GA) is used in this searching process, because it can return multiple regions of local optima. The regions of interest indicated by the GA can then be evaluated to obtain the true objective function value. With these updates, the surrogate model can be improved and the cycle can be repeated until a time limit is reached or, in the case of this investigation, a computational budget has been exhausted.

The presented benchmark procedure changes the number of variables and repeats each optimization a specified number of times to best judge the performance of a particular response-surface-based optimization strategy (in this case, the performance of five hyperparameter tuning strategies).

Figure 1 illustrates the main processes involved in the benchmarking procedure. The initial setup of the benchmark procedure requires a few key pieces of information: the design-of-experiment (DOE) size, the number of update cycles, and the maximum number of points in each cycle. Further information specific to the optimization strategy is also required, such as the method used to construct or tune the response surface and the method of searching the response surface.

The complete algorithm consists of three nested loops. The inner loop (denoted as loop 1 in Fig. 1) represents the typical activity of a designer employing a surrogate-based optimization. Results from an initial design of experiments are used to generate a response surface that is then searched to provide the specified number of update points. The variables from these update points are then used to carry out more computational experiments: in this case, computational fluid dynamics (CFD) simulations. The results from these numerical simulations are used to construct a new response surface that is searched again to obtain a further set of update points, if required.

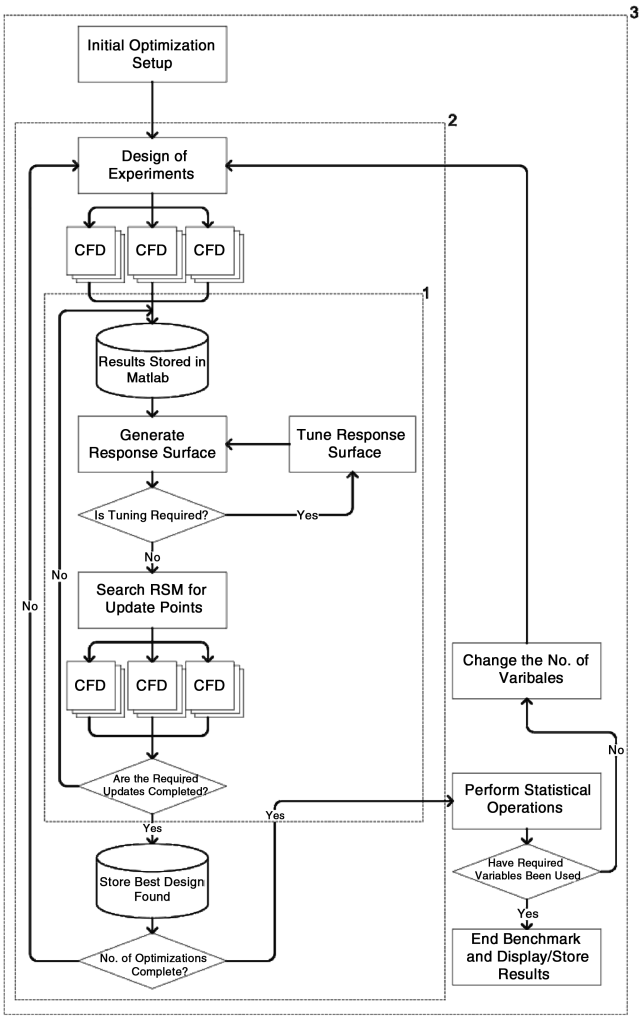


Fig. 1 Overview of the benchmark procedure.

This process is continued until a specified number of update cycles has been completed.

After storing the best design obtained, the entire optimization process (loop number 1) is then repeated using a DOE generated from a different random-number seed (loop 2 in Fig. 1). This prevents the "getting lucky" scenario whereby a point in the initial DOE falls on or near the global optimum. By carrying out a number of optimizations for the same problem (in terms of the number of variables), a statistical analysis of the results can be obtained that indicates the performance of the optimization methodology. The number of optimizations carried out depends on the available time for the benchmark. However, increasing the number of optimizations increases the confidence of the statistical analysis. The final outermost loop (loop 3 in Fig. 1) repeats the entire process but changes the number of variables that the optimization uses, thus assessing the ability of an optimization methodology to cope with problems of varying complexity.

The structure of this benchmark procedure allows it to be used to investigate many aspects of an optimization methodology (or, indeed, different optimization methodologies) and not just the performance of the tuning strategies studied here. Although an aerodynamic optimization problem is considered within this paper, the benchmark procedure could be just as easily applied to any appropriate optimization problem.

III. Overview of Kriging

Before considering the different tuning strategies within this paper, it is first necessary to consider the overall tuning process. The work of Jones [11] provides an extremely useful introduction to kriging, including the tuning and prediction processes, and a much more classical derivation of the process is presented in the original work of Sacks et al. [1].

The objective function values $y(\boldsymbol{x}_i)$ and $y(\boldsymbol{x}_j)$, which depend on the vectors of variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ of length d, will be close if the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is small. This can be modeled statistically by assuming that the correlation between two sets of random variables, $Y(\boldsymbol{x}_i)$ and $Y(\boldsymbol{x}_j)$, can be given by

$$\operatorname{corr}[Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j)] = \exp\left(-\sum_{\ell=1}^{d} \theta_\ell \|\boldsymbol{x}_{i\ell} - \boldsymbol{x}_{j\ell}\|^{p_\ell}\right) \tag{1}$$

where the hyperparameters θ_ℓ and p_ℓ determine the rate at which the correlation decreases and the degree of smoothness in the ℓth coordinate direction, respectively. A vector $\boldsymbol{y}$ consisting of a series of n objective function values can be considered:

$$\boldsymbol{y} = \begin{bmatrix} y(\boldsymbol{x}_1) \\ \vdots \\ y(\boldsymbol{x}_n) \end{bmatrix} \tag{2}$$

where the mean is $\mathbf{1}\hat{\beta}$, and $\mathbf{1}$ is a $n \times 1$ vector of ones. The covariance of $\boldsymbol{y}$ may then be written as

$$\operatorname{cov}(\boldsymbol{y}) = \sigma^2 \boldsymbol{R} \tag{3}$$

where σ^2 is the variance, and the elements of the matrix $\boldsymbol{R}$ are given by Eq. (1). Values of θ_ℓ and p_ℓ (the hyperparameters) are then chosen to maximize the likelihood on the observed data set $\boldsymbol{y}$. This maximum likelihood function is defined as

$$\frac{1}{(2\pi)^{\frac{n}{2}}(\sigma^2)^{\frac{n}{2}}|\boldsymbol{R}|^{\frac{1}{2}}} \exp\left[\frac{-(\boldsymbol{y}-\mathbf{1}\hat{\beta})^T \boldsymbol{R}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\beta})}{2\sigma^2}\right] \tag{4}$$

or, after taking the natural log of the function,

$$-\frac{n}{2}\ln(\sigma^2) - \frac{1}{2}\ln(|\boldsymbol{R}|) - \frac{(\boldsymbol{y}-\mathbf{1}\hat{\beta})^T \boldsymbol{R}^{-1}(\boldsymbol{y}-\mathbf{1}\hat{\beta})}{2\sigma^2} \tag{5}$$

Expressions for the optimal values of the mean

Table 1 Summary of tuning strategies

	Strategy	Description of tuning strategy
1	Heavy tune	5000-evaluation genetic algorithm and 5000-evaluation dynamic hill climb after the DOE and each update
2	Light tune	1000-evaluation genetic algorithm and 1000-evaluation dynamic hill climb after the DOE and each update
3	Single tune	5000-evaluation genetic algorithm and 5000-evaluation dynamic hill climb after the DOE only
4	Alternate tune	5000-evaluation genetic algorithm and 5000-evaluation dynamic hill climb after the DOE and alternate updates
5	θ tune	5000-evaluation genetic algorithm and 5000-evaluation dynamic hill climb after the DOE and each update, optimizing for a single common value of θ

$$\hat{\beta} = \frac{\mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}} \tag{6}$$

and variance

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\hat{\beta})^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\beta})}{n} \tag{7}$$

can be found by taking the relevant partial derivatives of the log-likelihood function and equating them to zero. Substituting the expressions for the optimal mean and variance into the log-likelihood function yields the concentrated likelihood function:

$$-\frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln(|\boldsymbol{R}|) \tag{8}$$

which is only dependent on the matrix $\boldsymbol{R}$ and hence on the hyperparameters that are tuned in an attempt to maximize the function. Hyperparameter tuning is therefore an optimization process in its own right.

Given the potentially noisy nature of many computational simulations that depend on discretization and iterative solutions, it is often extremely important to employ regression in the construction of a surrogate model [8]. Although kriging is often set up as an interpolating model, a regressing model can be constructed such that the sampled points no longer have an exact correlation with the resulting model. A constant λ is added to the diagonal of the correlation matrix $\boldsymbol{R}$, producing $\boldsymbol{R} + \lambda \boldsymbol{I}$. The magnitude of this constant is another hyperparameter varied in the tuning process, in which the optimal mean becomes

$$\hat{\beta} = \frac{\mathbf{1}^T (\boldsymbol{R} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y}}{\mathbf{1}^T (\boldsymbol{R} + \lambda \boldsymbol{I})^{-1} \mathbf{1}} \tag{9}$$

the variance is

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\hat{\beta})^T (\boldsymbol{R} + \lambda \boldsymbol{I})^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\beta})}{n} \tag{10}$$

and the concentrated likelihood function is given by

$$-\frac{n}{2}\ln(\hat{\sigma}^2) - \frac{1}{2}\ln(|\boldsymbol{R} + \lambda \boldsymbol{I}|) \tag{11}$$

Upon completion of the tuning process, the resulting surrogate model can be searched in an attempt to locate regions of optimal design.

It must be noted that the term hyperparameter refers only to θ_ℓ, p_ℓ, and λ that are employed in the mathematical description of the surrogate model used in the optimization process. The selection of their optimum values allows the surrogate model to better represent the true response of the objective function compared with a simpler surrogate model such as a radial basis function [10,11,13,14], which requires little or no tuning. The optimization process involved in hyperparameter tuning must not be confused with the search for a solution to the actual optimization problem (in this case, the inverse design of an airfoil). However, the variables of the optimization problem can be linked to the magnitude of the corresponding hyperparameters to determine a variable's importance during screening [16].

IV. Hyperparameter Tuning Strategies

A total of five different tuning strategies (summarized in Table 1) are investigated here using a low-fidelity inverse design problem. Each optimization strategy has the same overall structure. An initial DOE is performed on a set number of points. The resulting objective function values are then used to create a surrogate model. This model is then searched using a GA to find a series of update points that are evaluated in parallel using the CFD solver. The true objective function values at each update point are then used to improve the surrogate model.

The five strategies each employ a simplex search, followed by a GA and a dynamic hill climb [17] (DHC) to minimize the negative of the concentrated likelihood function [Eq. (11)]. The simplex search, using the Amoeba [18] algorithm, is started from a random set of hyperparameters and continued until one of two stopping criterion is reached: a predefined tolerance in the concentrated likelihood or $50d$ iterations have been completed. The hyperparameters defining the optimum point are then perturbed by up to 10% using a random number, and the simplex search repeated. The result of the simplex search is then included in the initial population of the GA, with the best design of the GA then used as the starting point of the DHC. Although a GA is a popular method of global optimization (and has been shown by Hollingsworth and Mavris [14] to be an effective and reliable method of tuning hyperparameters), the optimal point predicted can be inaccurate due to the GA's discretization of the design space. Although a GA can be relied upon to locate the region of an optimal design, it cannot be relied upon for an exact answer. A hill-climbing algorithm (in this case, the DHC) is therefore often used to refine the optima predicted by the GA.

Each of the five strategies differs in the degree of tuning used after each update. In both the heavy and light tuning strategies, the hyperparameters are tuned after the initial DOE and after every single set of updates to the model. The only difference is in the amount of tuning used in each case. The heavy tune consists of 5000 evaluations of the concentrated likelihood function using a GA (100 generations with a population size of 50), followed by a further 5000 evaluations using a DHC. The light tune uses only 1000 evaluations of each. The third strategy consists of a single heavy tune after the initial DOE followed by no further tuning after subsequent updates. The fourth strategy consists of a heavy tune after the initial DOE and after each alternate batch of updates. The fifth and final strategy involves the tuning of a single value of θ, which is assumed to be the same for every variable; the remaining hyperparameters are assumed to be constant ($p = 2$ and $\lambda = 10^{-6}$).

Two different computational budgets will also be used in the investigation, consisting of a total of 75 and 150 CFD evaluations. In accordance with the work of Sóbester et al. [19], each of these budgets has one-third of the evaluations reserved for the initial design of experiments, with the remainder used in the update process. A maximum of 10 CFD evaluations are permitted for each update cycle. The small-budget strategy therefore consists of a DOE of 25 evaluations, followed by five update cycles of up to 10 evaluations each. It must be noted that the budget of 10 CFD evaluations per update cycle may not be completely used in each cycle. The number of update points returned by the GA depends on the modality of the response surface generated; a highly modal surface may return the maximum 10 update points, whereas a surface with fewer undulations may return fewer update points. Therefore, the available budgets of 75 and 150 CFD evaluations may not be completely used in each optimization.

Table 2 Inverse design Hicks–Henne function parameter limits

Parameter	Lower limit	Upper limit
$A\ (x/c)$	−0.02	0.02
$x_p\ (x/c)$	0.01	0.95
t	1	6

The combination of different computational budgets and tuning strategies results in a total of 10 optimization strategies. Each strategy is used on a series of different optimization problems of varying complexity, achieved by altering the number of variables within the design problem. Each optimization is carried out a number of times with differing random-number seeds used to construct the Latin hypercube for the DOE each time. Because the random-number seeding is consistent for all of the tuning strategies, the CFD evaluations for each design of experiment only have to be calculated once. Each tuning strategy uses the same initial set of data to tune the initial surrogate model and thus reduces some of the computational expense of the investigation while providing a more meaningful comparison between the results of the different strategies.

V. Inverse Airfoil Design

The inverse design problem investigated here involves the modification of a baseline airfoil, the NACA 0012, through the addition of multiple Hicks–Henne [20] bump functions to the upper and lower airfoil surfaces in an attempt to recreate the surface pressure distribution of the RAE-2822 airfoil. A series of Hicks–Henne bump functions, described by

$$y = A[\sin(\pi x^{-\frac{\ln 2}{\ln x_p}})]^t \qquad x \in [0, 1] \tag{12}$$

(where the parameters A, x_p, and t denote the amplitude, position of the maximum, and sharpness of the bump, respectively) are applied to each surface through addition of the y coordinates. These Hicks–Henne control parameters form the variables in the optimization process and are allowed to vary according to the limits in Table 2. An example of a perturbation of the baseline geometry through addition of multiple Hicks–Henne functions is presented in Fig. 2.

Through manipulation of these parameters and through addition of multiple bump functions to the airfoil, the geometry parameterization can employ any number of variables. Increasing the number of variables increases the range of geometries that the parameterization can represent while simultaneously increasing the complexity of the optimization. When increasing the number of variables, additional bump functions are applied to the upper and lower surfaces of the baseline NACA-0012 airfoil in an alternate manner, with the upper-surface bump applied first. The parameters describing each Hicks–Henne function are designated as variables in a specific order. First, the amplitude of the function is defined as a variable, then the location of the maximum, followed by the sharpness. If only a proportion of the parameters are required to be designated as variables, the remaining parameters are kept constant with $x_p = 0.47$ and $t = 3.5$, corresponding to halfway between the limits of Table 2. A two-variable problem, for example, consists of a single bump function applied to the upper surface, with the parameters A and x_p permitted to vary and t equal to 3.5.

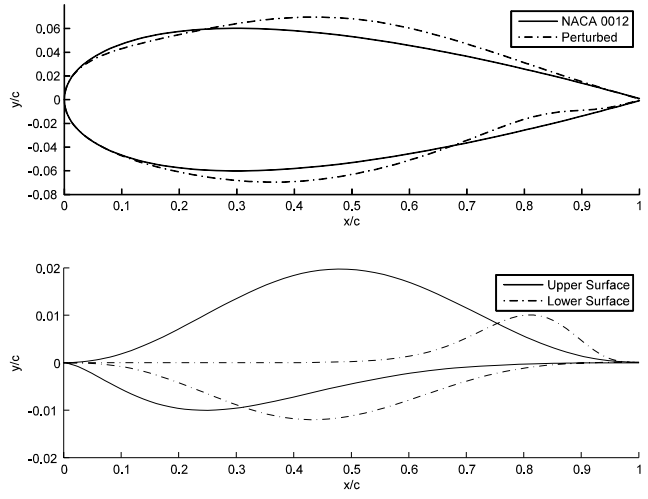


Fig. 2 An example of the NACA 0012 airfoil perturbed through the addition of four Hicks–Henne functions.

Identical airfoil geometries can be produced with a different set of variables, resulting in a design space containing multiple minima. Because the parameter limits are identical for each of the applied functions, it is possible to swap the parameters and obtain the same geometry. Consider the example airfoil in Fig. 2, in which two bumps were applied to the upper surface: the first bump subtracts a portion of the base airfoil from the leading edge, whereas the second bump thickens the airfoil at the midcamber point. The variables defining the first bump could be swapped with those of the second, resulting in an identical airfoil. The presence of multiple minima deliberately increases the difficulty of the optimization problem in a manner similar to other test functions, such as the Rastrigin function or the Keane bump function [21].

The parameter limits of each function permit a design produced using one parameterization to be recreated using a more complex parameterization. For example, any geometry resulting from the one-variable parameterization can be reproduced using the two-variable parameterization, because the default position of the maximum of the bump function can be replicated. The one-variable design space is therefore a line through the two-variable design space of constant maximum location ($x_p = 0.47$). The two-variable parameterization is itself a plane through the three-variable parameterization of constant sharpness ($t = 3.5$), and so forth. The continuity between the geometry parameterizations allows a useful comparison between the results of the benchmark optimizations.

The surface pressure distribution over each airfoil is calculated using the full potential code VGK (viscous Garabedian and Korn) [22] at an angle of attack of 2 deg, Mach 0.725, and a Reynolds number of 6.5×10^6. This pressure distribution is then compared with that of the RAE-2822 (also computed using VGK), and the rms error between the two distributions is calculated. The error between the pressure distributions then forms the objective function in the optimization process. The surrogate model therefore models the response of the error in pressure distribution to changes in the magnitude of the Hicks–Henne function parameters A, x_p, and t for each bump function applied to the baseline airfoil.

VI. Inverse Design Results and Discussion

A. Overview of Results

The inverse airfoil design problem was investigated using both of the computational budgets previously outlined for each of the five tuning strategies on a total of 14 increasingly complex geometry parameterizations. The parameterizations ranged from a simple one-variable problem to a difficult 30-variable problem in which the initial airfoil geometry was altered using a total of 10 Hicks–Henne functions. Each of the 14 optimization problems was carried out a total of 50 times, varying the Latin hypercube used to select the points in the DOE each time. This produced statistics with an associated high confidence level and mitigated the misleading effect of obtaining a near-optimum design with a point in the initial DOE.

The mean objective function values obtained for each tuning strategy, along with the standard deviation of the objective functions, are presented in the graphs of Fig. 3 for the 75-evaluation budget and in Fig. 4 for the 150-evaluation budget. The mean results of each optimization are presented together in Fig. 5 for ease of comparison.

The overall accuracy of the surrogate model used to find the final set of update points was assessed through the calculation of the r^2 correlation. The r^2 correlation compares the true objective function

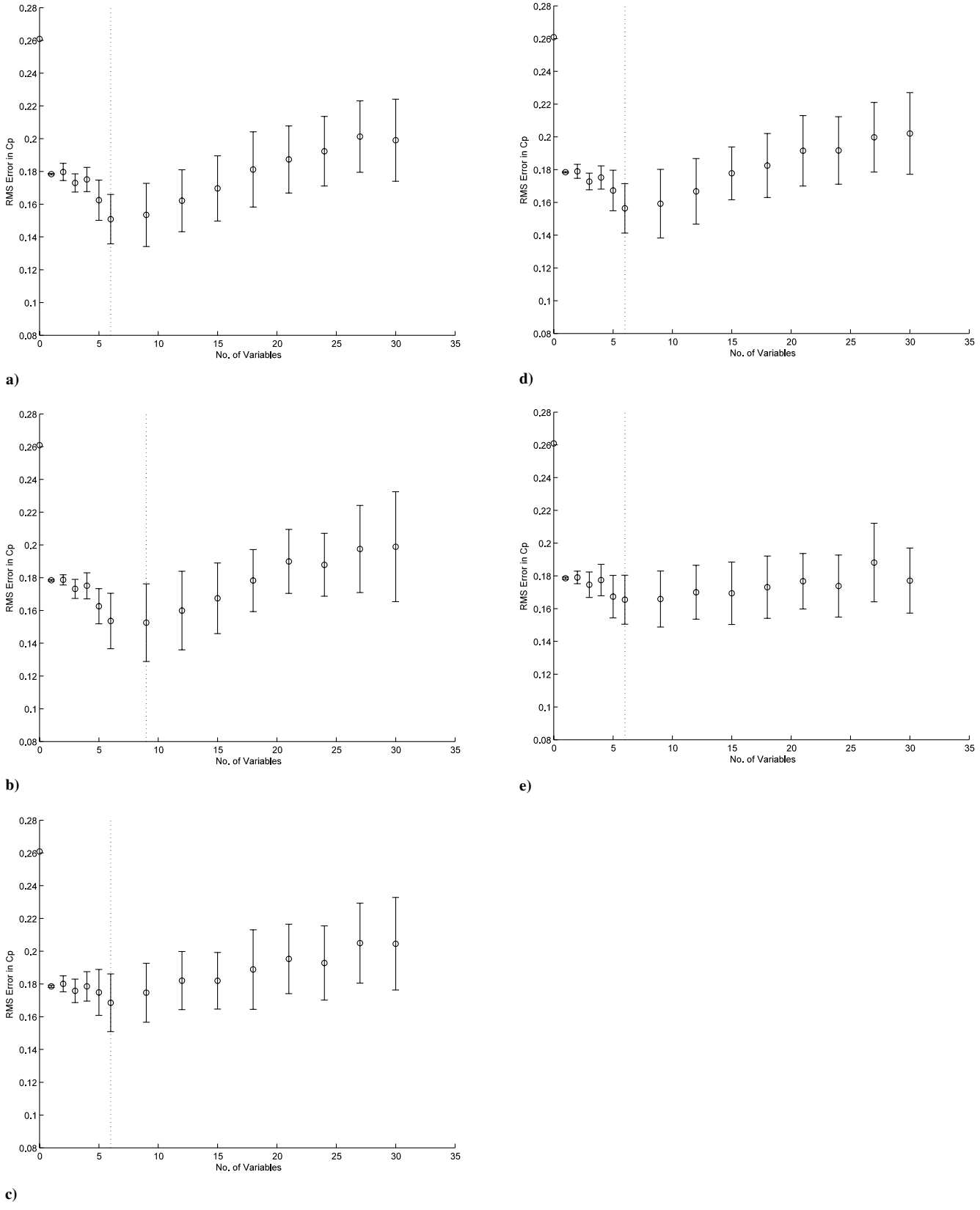


Fig. 3 Averaged results of the inverse design optimizations with a computational budget of 75 simulations using a) heavy, b) light, c) single, d) alternate, and e) θ tuning; the best result is indicated by a dotted line.

values at 250 design points with those predicted by the surrogate model. The average r^2 correlations for three different geometry parameterizations are presented in Table 3, with a value close to 1 indicating a high correlation between the true objective function and that predicted by the surrogate model.

B. Implications of Dimensionality

As expected, each of the five tuning strategies is adversely affected by the increase in dimensionality of the design problem as the number of variables used to define the geometry parameterization increases. Given a fixed simulation budget, as the dimensionality of

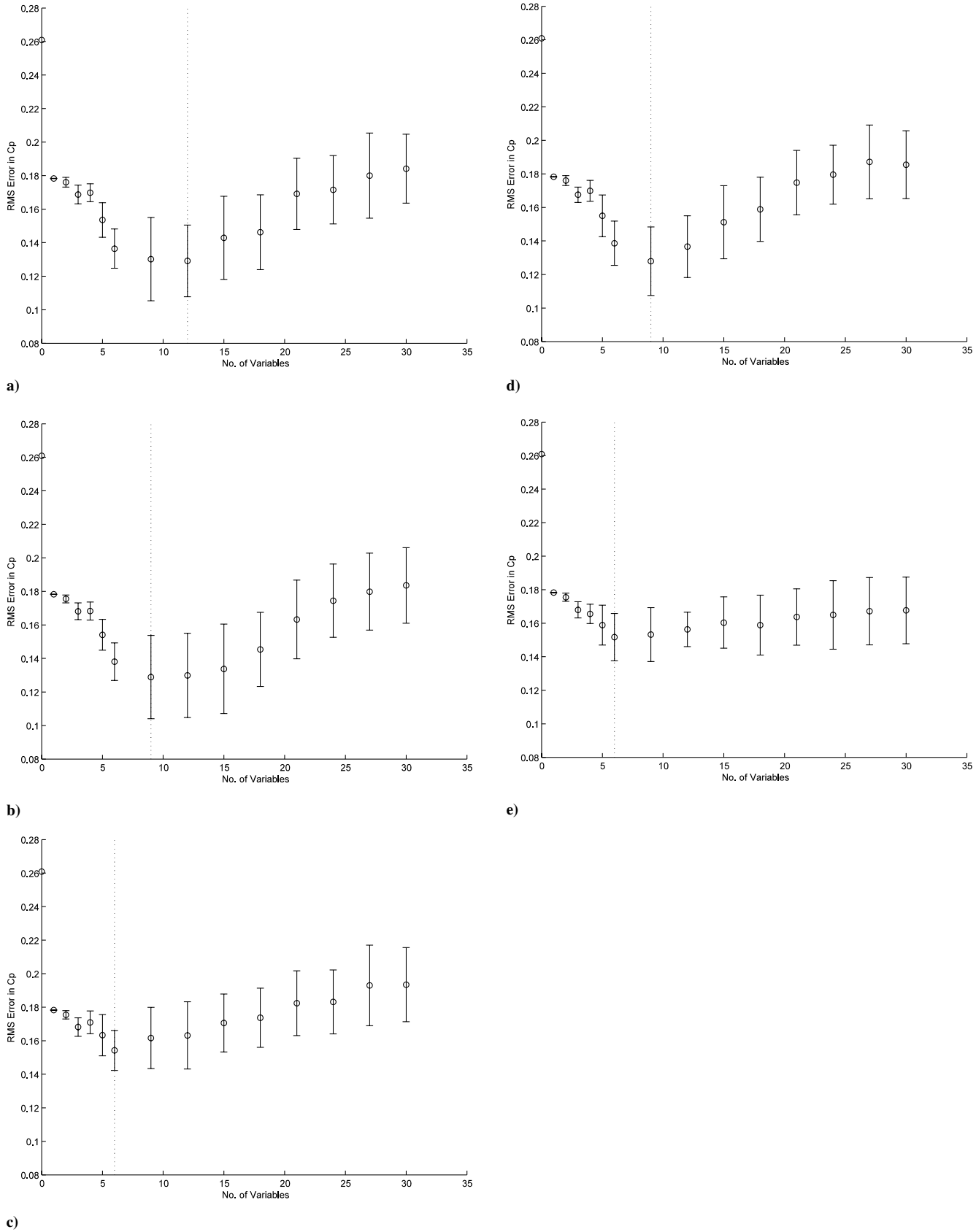


Fig. 4 Averaged results of the inverse design optimizations with a computational budget of 150 simulations using a) heavy, b) light, c) single, d) alternate, and e) θ tuning; the best result is indicated by a dotted line.

the problem increases, the optimization processes are capable of finding increasingly better designs until the dimensionality reaches a certain point. Beyond this threshold (indicated by the dotted lines in Figs. 3 and 4), the problem becomes too complex for the given optimization strategy to adequately search. Therefore, as the complexity of the design problem increases further, the optimization strategy generally produces best designs, with objective functions worse than designs obtained with less complex parameterizations.

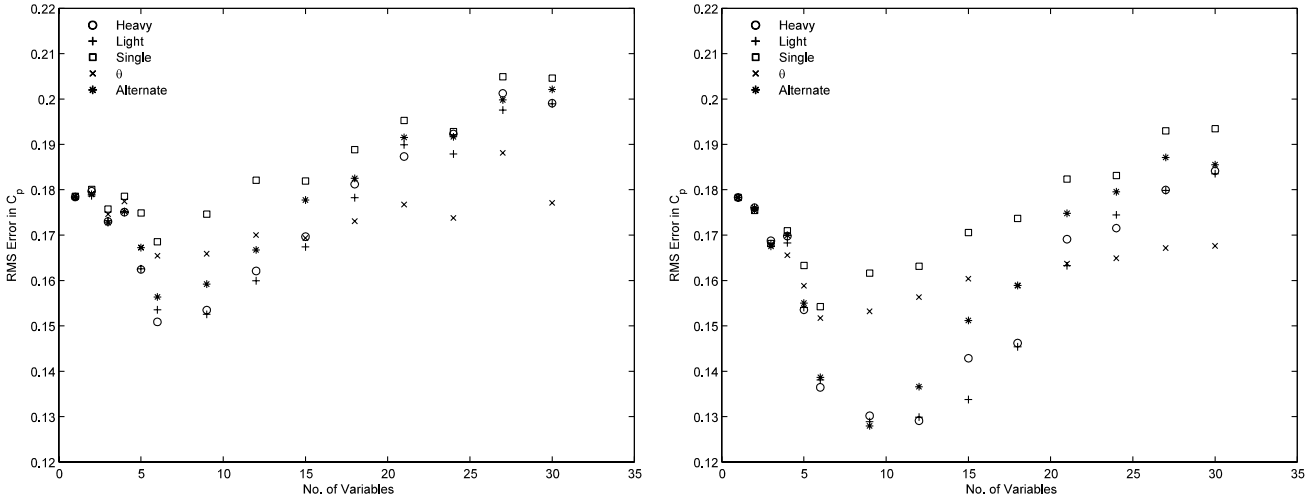


Fig. 5 Average objective function values obtained for each of the five tuning strategies for problems of varying complexity using a budget of a) 75 simulations and b) 150 simulations.

Consider the heavy tuning strategy with the small simulation budget (Fig. 3a): As the complexity of the geometry parameterization increases from one to six variables, the strategy is able to improve the mean objective function value. The additional flexibility introduced by the increased number of variables allows the target pressure distribution of the RAE-2822 airfoil to be more closely attained. When the parameterization consists of more than six variables, the given budget is no longer sufficient to search the design space, resulting in increasingly suboptimal best designs. Because of the continuity between the geometry parameterizations, even if the increase in complexity of the parameterization produces a design that offers no improvement over a simpler parameterization, an optimization should, at the very least, produce an identical design. The inability of the optimization strategy to regularly obtain this design is therefore a direct result of the increase in the dimensionality of the problem.

Increasing the size of the computational budget, as seen in Figs. 4 and 5, has the effect of delaying this degradation in performance. The heavy tuning strategy with the 150-simulation budget generally continues to improve the mean objective function up to a 12-variable design problem, after which the performance of this strategy is also degraded by increasing dimensionality.

The effect of the simulation budget on the search for an optimum design is further emphasized in Figs. 6 and 7, which show the best overall design obtained using both of the simulation budgets for the three- and nine-variable problems, respectively. Both simulation budgets produce similar designs for the three-variable problem, although the design resulting from the larger budget is marginally better, giving a rms error of 0.1608, compared with 0.1622 for the smaller budget. Observe that the upper-surface pressure distribution upstream of the shock wave obtained using the larger simulation is very slightly closer to the target pressure.

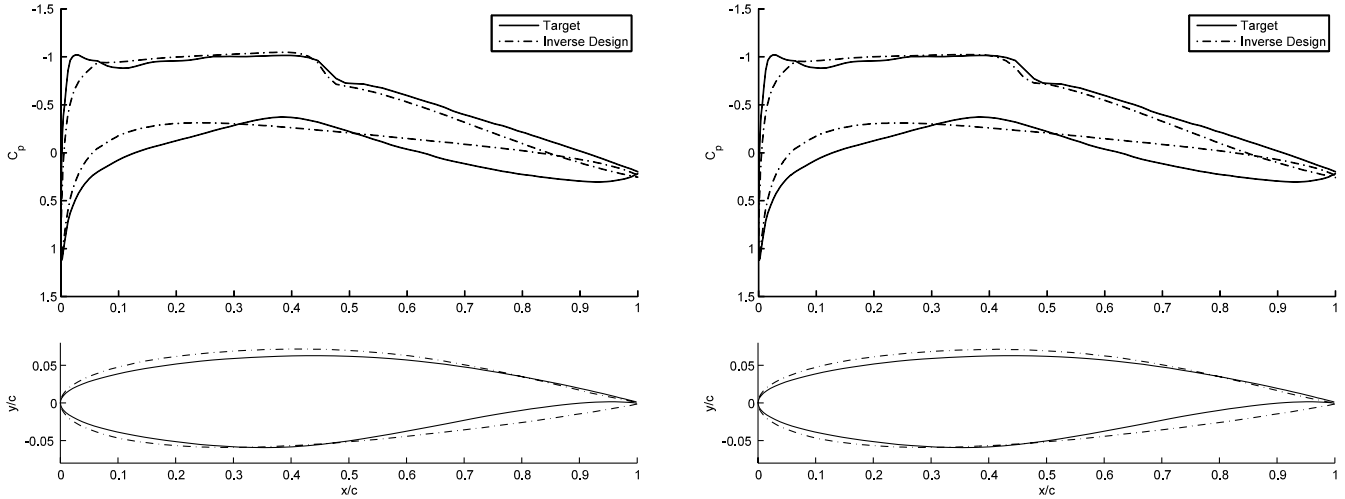
When the complexity of the optimization problem is increased to nine variables, consisting of two Hicks–Henne functions on the upper surface and one on the lower surface, the effect of the computational budget is much more apparent. There is a significantly greater difference in the best designs, with the larger budget producing a rms error of 0.076 and the smaller budget producing an error of 0.1138. This difference is demonstrated graphically in Fig. 7. The best design obtained with the large budget produces a pressure distribution over the upper surface that matches the target pressure much more closely; only the region around the shock wave and the suction peak over the leading edge fail to be accurately reproduced. The lower-surface pressures differ significantly from the target, with the exception of a region close to the trailing edge of the geometry optimized using the larger simulation budget. This inaccuracy is mainly due to the inadequacies of the lower-surface geometry parameterization; the application of a single bump function to the lower surface of the baseline airfoil allows the lower-surface geometry to be adjusted locally in only one region. To obtain a more accurate reproduction of the RAE-2822 airfoil requires the modification of the baseline airfoil close to the leading and trailing edges simultaneously, with little modification in between. Introducing a second bump to the lower surface reduces the objective function further, as one can observe in Fig. 4a. However, the increased complexity of the design problem reduces the ability of the optimization to search effectively, resulting in only a slight improvement over the nine-variable problem.

The average r^2 correlations presented in Table 3 verify the positive impact of increasing the size of the simulation budget on the overall accuracy of the surrogate model. These results also reinforce the negative effect of increasing problem dimensionality given a fixed simulation budget. As the dimensionality of the optimization problem increases, the available budget becomes increasingly incapable of accurately representing the global response of the objective function (demonstrated by the decrease in the average r^2 correlation).

Given a limited simulation budget, variable-screening techniques could be used to reduce the complexity of an optimization problem. There are a number of different variable-screening methodologies

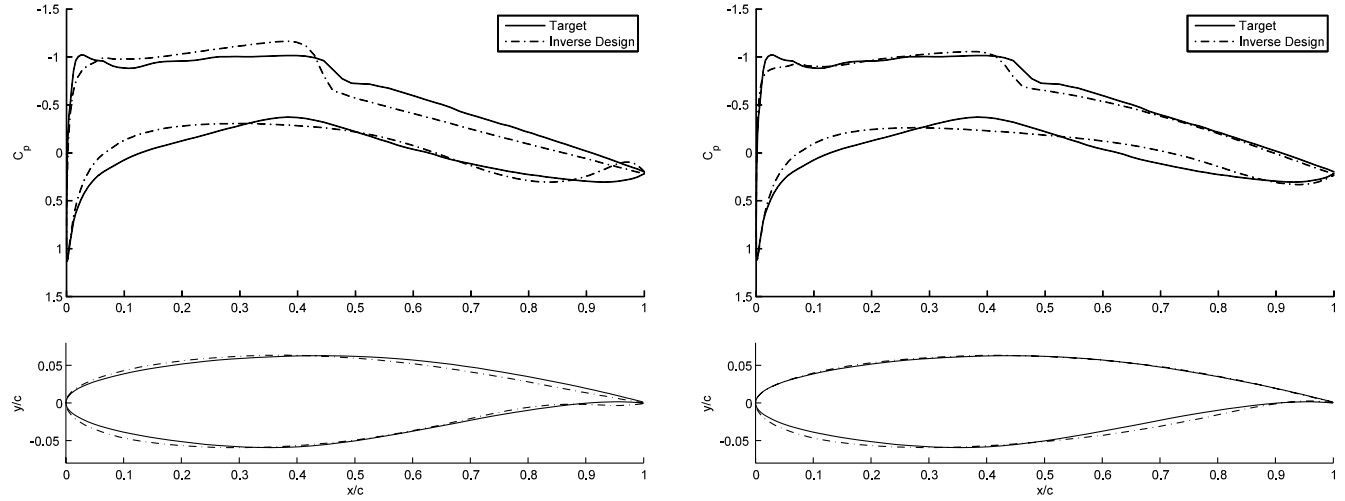
Table 3 Average r^2 correlation of the final surrogate model for three geometry parameterizations

	75 Simulation budget			150 Simulation budget		
Tuning strategy	3 variables	6 variables	9 variables	3 variables	6 variables	9 variables
Heavy tune	0.731	0.536	0.125	0.854	0.715	0.257
Light tune	0.760	0.563	0.145	0.852	0.708	0.228
Single tune	0.568	0.140	0.013	0.826	0.335	0.024
Alternate tune	0.755	0.556	0.118	0.854	0.716	0.250
θ tune	0.609	0.214	0.077	0.730	0.327	0.108



a) b)

Fig. 6 Best airfoil design obtained with the three-variable optimization using the a) 75-simulation budget and b) 150-simulation budget.



a) b)

Fig. 7 Best airfoil design obtained with the nine-variable optimization using the a) 75-simulation budget and b) 150-simulation budget.

within the literature; Welch et al. [16] for example, studied the influence of the hyperparameters of each variable on the concentrated log-likelihood to determine the most active variables, and Morris [23] used an estimation of the partial derivatives of the objective function. All of the variable-screening techniques involve using a proportion of the total simulation budget to determine the most important variables within the problem. The remaining simulation budget can be employed in the actual optimization of the problem using this reduced set of variables. It must be noted, however, that a reduction in variables reduces the flexibility of a parameterization to obtain an optimum design while reducing the simulation budget available to achieve an optimum using the variable subset. Although such techniques could be used in conjunction with the current inverse design problem, the complex interplay between the proportion of the budget spent on variable screening while achieving a good design with the reduced variable set is deemed beyond the scope of this paper.

C. Performance of the Heavy and Light Tuning Strategies

The results presented in Figs. 3–5 demonstrate only a marginal difference between the heavy and light tuning strategies for those optimizations in which the simulation budget could be considered adequate to search the design space (i.e., up to the six-variable problem for the small budget and the 12-variable problem for the large budget). The mean objective functions (and even the standard deviations) are approximately equal, and the r^2 correlation results (Table 3) indicate little difference in the global accuracy of the surrogate models produced using both tuning strategies. This indicates that the hyperparameters obtained after tuning are similar, and, as a result, the optimization process searches similar response surfaces. A similar response surface means that the results of each optimization, and hence the resulting statistics, closely correspond. The presented results therefore indicate that the light tuning strategy could be considered sufficient to tune the hyperparameters of the response surfaces constructed for these optimizations.

D. Performance of the Single Tune Strategy

The results of very low-dimensional problems differ only marginally from the results of the heavy and light tuning strategies when the hyperparameters are tuned only once after the initial design of experiments. However, this difference grows substantially as the complexity of the optimization problem increases. The mean objective function obtained for the 12-variable problem using the single tuning strategy and large budget is approximately 25% worse than that obtained with the heavy tuning strategy. The performance of the single tuning strategy, as with each of the other tuning

Table 4 Comparison of different DOE budgets on the search for an optimum design using the single tune strategy

	Ratio of DOE size to total simulation budget	30/150	50/150	80/150	100/150	130/150
6 variables	Mean objective function	0.1656	0.1542	**0.1501**	0.1530	0.1559
	Standard deviation	0.0153	0.0119	0.0137	**0.0106**	0.0124
9 variables	Mean objective function	0.1689	0.1616	0.1550	**0.1509**	0.1517
	Standard deviation	0.0171	0.0183	0.0179	0.0162	**0.0153**
12 variables	Mean objective function	0.1773	0.1632	0.1639	**0.1597**	0.1624
	Standard deviation	0.0192	0.0200	0.0189	0.0188	**0.0135**

strategies, is improved to some degree with the increase of the simulation budget, as shown in Figs. 3c and 4c.

The ability of the single tuning strategy to produce results comparable with those of the heavy and light tuning strategies for simple problems is due to the ability of the design of experiments to adequately seed the design space. The number of points in the initial DOE is such that the hyperparameters resulting from the initial tuning process accurately capture the correlation between points and the trends of the true response. This is confirmed when one considers the r^2 correlation results for the three-variable problem, which are reasonably close to those of the heavy and light tuning strategies. With an initial accurate set of hyperparameters, the global exploration of the response surface can therefore find basins of optimal design easily. When the complexity of the design problem is increased, the number of points in the design of experiments cannot produce such an accurate representation of the true response surface (again confirmed by the results in Table 3). Subsequent update points are essentially wasted, because without further tuning of the hyperparameters the correlation, smoothness (and even the degree of regression used in the construction of the response surface) cannot be updated and corrected. Consequently, the genetic algorithm finds basins of optima in an inaccurate response surface, and the updates are unable to improve upon the current best design. The best designs of such an optimization are therefore worse than those in which the hyperparameters are continually reassessed.

The improvement of the single tune strategy as the simulation budget is increased is therefore a result of the greater number of points making up the initial design of experiments. Table 3 indicates that the inclusion of a larger number of points results in a more accurate set of hyperparameters from the initial tune, producing a better representation of the true surface.

Given the increased performance of the optimization strategy with an increase in the size of the design of experiments, it stands to reason that by altering the ratio of the total simulation budget used in the initial DOE, the performance of this tuning strategy can be improved. In addition to the results already presented, three of the design problems were chosen and optimized using four additional DOE sizes. Table 4 presents the results of this additional investigation, including the original results using the DOE simulation budget of 50 for comparison. The presented results were once again averaged over 50 optimizations.

The results presented in Table 4 demonstrate that a reduction in the number of DOE points results in a degradation in performance, whereas increasing the number of DOE points increases the performance of the tuning strategy for each of the problems up to a point. The larger design of experiments produces a more accurate response surface, allowing the update points to be chosen more effectively, resulting in a better design. There is, however, a tradeoff between exploration and exploitation: as the size of the design of experiments is increased, a smaller proportion of the budget is available to exploit any potential regions of optimal design. For example, the six-variable optimization sees a deterioration in performance when more than 80 points are used in the DOE. Even though the hyperparameters resulting from the tuning process produce a response surface model that more accurately predicts the true response, there is an insufficient budget of simulations available to exploit it.

The effect of DOE size is shown graphically by the optimization histories presented in Fig. 8. The smallest (30-point) design-of-experiment results in a set of hyperparameters and corresponding response surface that generate update points that improve very little upon the best design obtained in the DOE in the majority of cases. This can be quantified if one considers the mean improvement of 0.0206, where the mean improvement is the difference between the objective function of the best design found in the DOE and the best design found after all of the updates, averaged over 50 optimizations. Increasing the size of the design of experiments produces a more effective response surface, and the updates can improve more upon the best design (Fig. 8b with a mean improvement of 0.0291). As the size of the design of experiments is increased further still, the remaining simulation budget available for updates is insufficient to fully exploit the response surface and improve designs (Figs. 8c and 8d). The mean improvement upon the best design of the DOE mirrors this, decreasing from 0.0234 to 0.0153, as the ability of the optimization to exploit the response surface diminishes.

In addition to the effect of the DOE size relative to the total available budget, the complexity of the problem has an impact on the optimum number of points in the design of experiments. The designs obtained using the 12-variable geometry parameterization continue to improve with 100 points in the DOE (compared with the 80-point DOE in the six-variable problem and the 100-point DOE in the nine-variable problem).

The optimization histories of Fig. 9 illustrate another interesting advantage of the heavy tuning strategy: the reduction in the total number of simulations. Per the previous description of the update strategy, depending on the modality of the response surface, the maximum budget of 10 update points may not be used in every update cycle. Hence, the continual tuning of the hyperparameters resulting in smoother response surfaces enables the available updates to be used more effectively, reducing the simulation cost of the optimization process. These unused updates could be employed in further exploitation of the metamodel of the heavy tuning strategy, possibly resulting in better designs.

The gain in performance associated with increasing the size of the design of experiments (shown in Table 4) is insignificant when one considers that the average objective function obtained using the heavy tuning strategy is 0.136, 0.130, and 0.129 for the 6-, 9-, and 12-variable problems, respectively. One can surmise that an optimization using the single tune strategy requires a larger simulation budget to perform as well as a smaller budget with continuous hyperparameter tuning. Considering the small cost of the airfoil simulations used in the current optimizations, increasing the budget in such a manner would have little effect compared with the cost of the hyperparameter tuning. However, when an optimization involves high-fidelity 3-D computational simulations requiring many hours of run time, it may be infeasible to significantly increase the computational budget. Nonetheless, if the cost of tuning the hyperparameters is comparable with that of the individual simulations, a balance may have to be struck between the total number of simulations and the level of tuning.

E. Performance of an Alternate Tuning Strategy

Tuning the hyperparameters after alternate updates offers a compromise between the single tune strategy and the heavy tuning strategy. The total amount of tuning carried out is significantly reduced, which offers a reduction in the total tuning time per optimization. Because the hyperparameters are reassessed throughout the updating process, although not as often as with the heavy tuning strategy, the response surface is better able to adjust to the

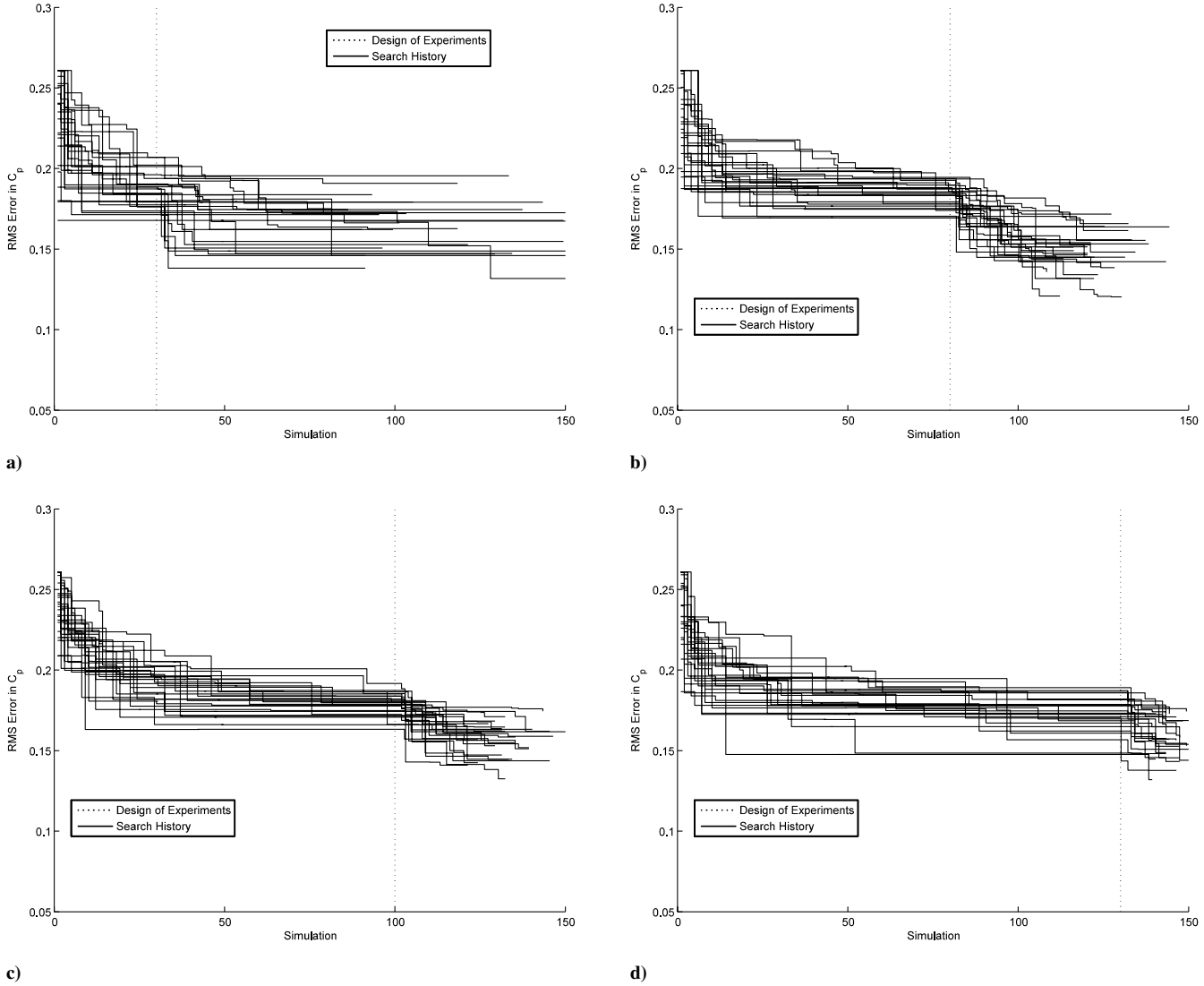


Fig. 8 Optimization histories for the six-variable problem using the single tune strategy and a) 30, b) 80, c) 100, and d) 130 points in the initial DOE.

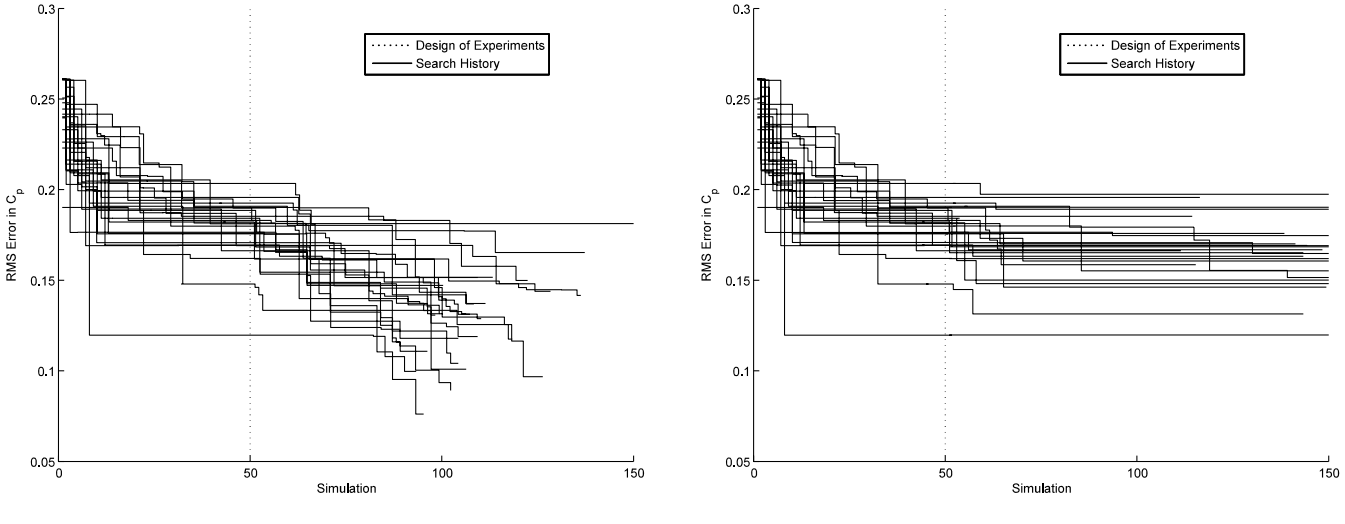


Fig. 9 Optimization search histories of the nine-variable design problem using the 150-simulation budget and the a) heavy tuning strategy and b) single tune strategy.

potential lack of accuracy in the hyperparameters obtained after tuning the response surface based on the initial design of experiments. This reassessment of hyperparameters enables the optimization strategy to more effectively exploit regions of optimal design and, as such, produces better designs compared with the single tune strategy.

The increase in performance over the single tune strategy is clearly demonstrated by the results shown in Fig. 5. Other than the light tuning strategy, the alternate tuning consistently produces results close to that of the heavy tuning strategy for those optimizations in which the simulation budget can be deemed sufficient. Degradation in performance of the strategy with respect to the heavy tuning strategy occurs when the dimensionality of the design problem becomes an issue. Using the small simulation budget, the alternate tuning strategy produces designs with objective functions close to those obtained using the heavy tuning strategy for the first four design problems. After this point, although there is a continual reduction in the mean objective function of the designs for the five- and six-variable problems, the reduction is not as significant as that obtained with the heavy tuning strategy. The results of the optimizations using the larger simulation budget remain close to those of the heavy tuning strategy up to the nine-variable problem. The performance of the alternate strategy degrades after that point, whereas the heavy tuning strategy produces a slight improvement of the objective function.

The alternate reassessment of the hyperparameters results in a surrogate model with an overall accuracy similar to that of the heavy and light tuning strategies. This is confirmed when one considers the average r^2 correlation of the surrogate models, which are consistently close to those of the heavy and light tuning strategies. Alternate tuning of hyperparameters could therefore be considered to perform as well as the heavy and light tuning strategies when the dimensionality of the problem with regard to the simulation budget is not an overriding issue.

The alternate tuning strategy could be improved further when one considers the previous comparison of the heavy and light tuning strategies. The results of this comparison indicate that the 5000 concentrated likelihood function evaluations of the genetic algorithm and dynamic hill climb used in the alternate tuning could be reduced with minimal loss of performance. Reducing the number of function evaluations in the alternate tuning strategy would produce an optimization strategy competitive with that of the single tune strategy or θ tuning strategy in terms of total tuning time, but with the potential to produce better designs.

F. Performance of the Single θ Tuning Strategy

Tuning a single common hyperparameter produces results very similar to those obtained using the heavy tune when a single-variable parameterization is optimized. As the number of variables used in the geometry parameterization is increased, the results of each optimization consistently falls short of that of the heavy, light, and alternate tuning strategies when the simulation budget is sufficient: that is, up to the six- and nine-variable design problems for the small and large simulation budgets, respectively. The results, however, are never as poor as those obtained through the single tune strategy.

When one compares the optimization history of the nine-variable design problem presented in Fig. 10 with that of the single tune strategy (Fig. 9b), one can see a continual improvement of the majority of the optimizations after the initial design of experiments. Although the single tune strategy offers little improvement over the best design obtained in the design of experiments, the single θ tuning strategy's continual reassessment of the hyperparameters after each update allows subsequent updates to continue to reduce the rms error in the pressure distributions. This improvement in objective function occurs even though the global accuracy of the surrogate model may be considerably less than that obtained using the single tune strategy (Table 3).

Tuning a single hyperparameter has a smoothing effect on the response surface, removing the significance of each individual design variable to the objective function. This smoothing is apparent when one compares the optimization history of Fig. 10 with that of the heavy tuning strategy presented in Fig. 9a. The heavy tuning strategy uses a much larger proportion of the available simulation budget (i.e., closer to the full allocation of 10 evaluations per update cycle) than the single θ tuning strategy. The process of searching the design space for potential regions of interest is therefore producing far more update points, indicating that the response surface produced with the heavy tuning strategy has more regions of minima to exploit than the surface obtained using the single θ strategy. Given the same initial design of experiments, this reduction in the number of returned update points is likely to result from the smoothing effect of using a single set of hyperparameters.

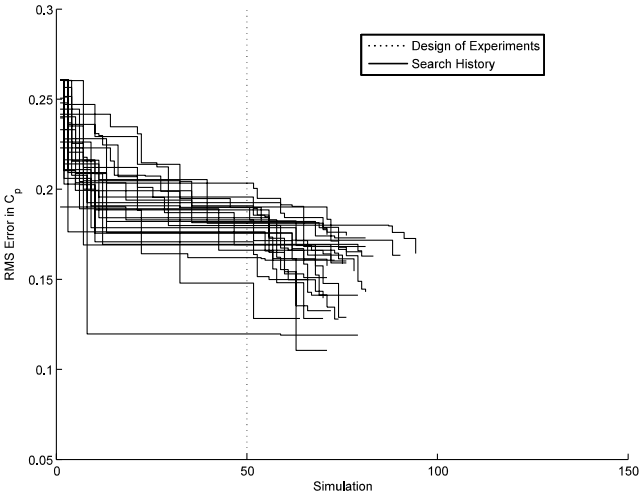


Fig. 10 Optimization search history of the nine-variable design problem using the single θ tuning strategy using the 150-simulation budget.

Increasing the dimensionality of the design problem beyond the apparent limit of the simulation budget (indicated by the dotted line in Figs. 3 and 4e does not produce the same drop in performance that is observed with the other tuning strategies. Considering the most difficult optimization, that of the 30-variable design problem using the 75-simulation budget, there is a large difference in the objective functions obtained using the heavy tuning strategy and the single θ tuning strategy. Tuning each of the 61 possible hyperparameters individually is an extremely complex optimization problem, one for which even the heavy tuning strategy may be insufficient. The optimization histories for the 30-variable design problem, presented in Fig. 11, reinforce the failings of the heavy tuning strategy in this regard. The update points chosen using the metamodel resulting from a heavy tune improve very little upon the best design obtained using the design of experiments for the majority of optimizations. The single θ strategy, on the other hand, can improve on each of the designs. This is confirmed when one compares the average improvement in the objective function of the best design of the DOE with the final best design. The heavy tuning strategy has an average improvement of 0.015, whereas the θ tuning strategy has a much greater average improvement of 0.0372. These results demonstrate the importance of an appropriate set of hyperparameters in an optimization strategy; that is, a small set of well-tuned hyperparameters may outperform a larger more complex set that is inaccurately tuned. Of course, for the problem considered here, all of the variables have broadly similar importance on the problem, and so using a single value of θ is plausible. In cases in which the design variables differ significantly in nature and importance, this might well not be the case.

Finally, it should be noted that, in a manner similar to the alternate tuning strategy, the tuning effort used in the single θ strategy could be reduced by decreasing the number of iterations of the genetic algorithm and dynamic hill climb, with little loss in performance.

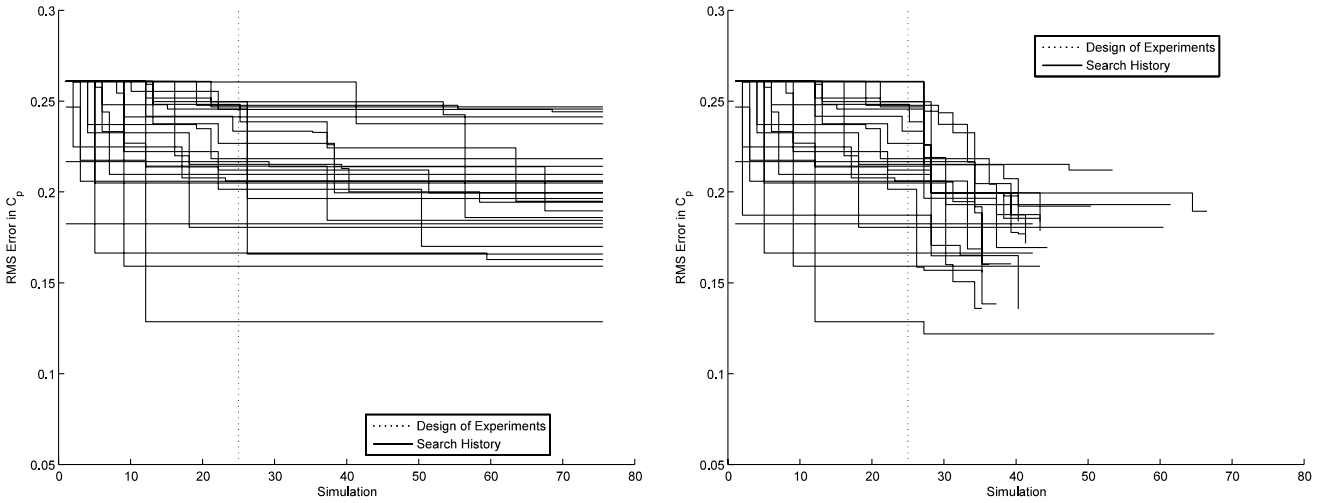


Fig. 11 Optimization search histories of the 30-variable design problem using the a) heavy tuning strategy and b) single θ tuning strategy using the 75-simulation budget.

VII. Conclusions

The inverse design of an airfoil was used to compare the performance of five kriging hyperparameter tuning strategies that were employed to construct and update a surrogate model of the aerodynamic response to geometric variation. Two of the strategies involve different degrees of tuning: after the design of experiments and every update cycle, designated as the heavy and light tuning strategies. Tuning after only the design of experiments is considered, as is tuning after alternate updates and tuning a single pair of hyperparameters after every update. The inverse design problem employs multiple Hicks–Henne functions to produce geometry parameterizations of varying complexity while maintaining a level of continuity between the different parameterizations.

All of the tuning strategies demonstrate the negative impact of increasing dimensionality on an optimization given a fixed simulation budget: in particular, the increasing inability of an optimization to produce good designs.

The results of the heavy and light tuning strategies displayed little difference for those optimizations with an adequate simulation budget for the problem (i.e., for problems with fewer than six variables for the small simulation budget and with fewer than 12 variables for the large simulation budget). The use of the light tuning strategy could therefore be recommended over the heavy tuning strategy for such problems.

Tuning the hyperparameters of the metamodel only once after the design of experiments was demonstrated to perform extremely poorly compared with the other tuning strategies considered. Only with some form of continual reassessment of the hyperparameters throughout the updating process is the available computational budget used effectively. To produce results on a par with that of the other strategies, it was demonstrated that a larger computational budget, in terms of both the initial design of experiments and updates, is required.

Tuning the hyperparameters after alternate updates produces results comparable with those of the heavy and light tuning strategies when problem dimensionality, in terms of the available simulation budget, is not an issue. This particular tuning strategy offers significant savings over the computational effort required in tuning the hyperparameters after every update, while having little impact on the optimizations ability to find good designs.

Tuning a single common hyperparameter (i.e., the same θ for every variable) is demonstrated to perform poorly compared with the other strategies when the available simulation budget is sufficient for the problem. However, when problems of high-dimensionality with limited simulation budget are considered, the tuning of a reduced set of hyperparameters produces a surprising improvement on the results obtained using the more expensive heavy tuning strategy. This indicates that given a high-dimensional problem in which extensive tuning of the complete set of hyperparameters is prohibitively expensive, extensively tuning a reduced set of hyperparameters may outperform an inaccurately tuned but complete set of hyperparameters. This final conclusion will depend, however, on the degree of variation in the problem variables.

Acknowledgments

This work was undertaken as part of the wider Centre for Fluid Mechanics Simulation (CFMS) framework, aimed at delivering a paradigm shift in the capability of fluid mechanics simulation systems. This framework was established to manage a sustained program of research projects, both private venture and government-supported. More details can be found on www.cfms.org.uk. The encouragement and advice of T. Barrett, A. Forrester, and A. Sóbester of the School of Engineering Sciences, Southampton University, is also greatly appreciated.

References

[1] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435. doi:10.1214/ss/1177012413

[2] Hoyle, N., Bressloff, N. W., and Keane, A. J., "Design Optimization of a Two-Dimensional Subsonic Engine Air Intake," *AIAA Journal*, Vol. 44, No. 11, 2006, pp. 2672–2681. doi:10.2514/1.16123

[3] Forrester, A. I. J., Bressloff, N. W., and Keane, A. J., "Optimization Using Surrogate Models and Partially Converged Computational Fluid Dynamics Simulations," *Proceedings of the Royal Society of London A*, Vol. 462, No. 2071, 2006, pp. 2177–2204. doi:10.1098/rspa.2006.1679

[4] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A., "Sequential Kriging Optimization Using Multiple-Fidelity Evaluations," *Structural and Multidisciplinary Optimization*, Vol. 32, No. 5, 2006, pp. 369–382. doi:10.1007/s00158-005-0587-0

[5] Sakata, S., Ashida, F., and Zako, M., "Structural Optimization Using Kriging Approximation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, Nos. 7–9, 2003, pp. 923–939. doi:10.1016/S0045-7825(02)00617-5

[6] D'Angelo, S., and Minisci, E. A., "Multi-Objective Evolutionary Optimzation of Subsonice Airfoils by Kriging Approximation and Evolution Control," *2005 IEEE Congress on Evolutionary Computation*, Vol. 2, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2005, pp. 1262–1267.

[7] Keane, A. J., "Statistical Improvement Criteria for Use in Multiobjective Design Optimization," *AIAA Journal*, Vol. 44, No. 4, 2006, pp. 879–891.

[8] Forrester, A. I. J., Keane, A. J., and Bressloff, N. W., "Design and Analysis of 'Noisy' Computer Experiments," *AIAA Journal*, Vol. 44, No. 10, 2006, pp. 2331–2339.
doi:10.2514/1.20068

[9] Kennedy, M. C., and O'Hagan, A., "Predicting the Output from a Complex Computer Code When Fast Approximations are Available," *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13.
doi:10.1093/biomet/87.1.1

[10] Jin, R., Chen, W., and Simpson, T. W., "Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria," *Structural and Multidisciplinary Optimization*, Vol. 23, No. 1, 2001, pp. 1–13.
doi:10.1007/s00158-001-0160-4

[11] Jones, D. R., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 345–383.
doi:10.1023/A:1012771025575

[12] Won, K. S., and Ray, T., "Performance of Kriging and Cokriging Based Surrogate Models Within the Unified Framework for Surrogate Assisted Optimization," *Proceedings of the 2004 Congress on Evolutionary Computation*, Vol. 2, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 2004, pp. 1577–1585.

[13] Simpson, T. W., Peplinski, J. D., and Kock, P. N., "Metamodels for Computer-Based Engineering Design: Survey and Recommendations," *Engineering with Computers*, Vol. 17, No. 2, 2001, pp. 129–150.
doi:10.1007/PL00007198

[14] Hollingsworth, P. M., and Mavris, D. N., "Gaussian Process Meta-Modeling: Comparison of Gaussian Process Training Methods," AIAA 3rd Annual Aviation Technology, Integration and Operations (ATIO) Tech, Denver, CO, AIAA Paper 2003-6761, 2003.

[15] Pound, G., and Price, A., "The Geodise OptionsMatlab Toolbox—A User's Guide," http://www.geodise.org/documentation/OptionsMatlab/html/index.htm.

[16] Welch, W. J., Buck, R. J., Sacks, J., and Wynn, H. P., "Screening, Predicting and Computer Experiments," *Technometrics*, Vol. 34, No. 1, 1992, pp. 15–25.
doi:10.2307/1269548

[17] Yuret, D., and Maza, M., "Dynamic Hill Climbing: Overcoming the Limitations of Optimization Techniques," *Proceedings of the Second Turkish Symposium on Artificial Intelligence and Neural Networks*, 1993, pp. 201–212.

[18] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes*, 3rd ed., Cambridge Univ. Press, Cambridge, England, U.K., 1986.

[19] Sóbester, A., Leary, S. J., and Keane, A. J., "On the Design of Optimization Strategies Based on Global Response Surface Approximation Models," *Journal of Global Optimization*, Vol. 33, No. 1, 2005, pp. 31–59.
doi:10.1007/s10898-004-6733-1

[20] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[21] Keane, A. J., and Nair, P. B., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, 1st ed., Wiley, Chichester, England, U.K., 2005.

[22] Freestone, M. M., "VGK Method for Two-Dimensional Aerofoil Sections," Engineering Sciences Data Unit, Rept. 96028, London, 1996.

[23] Morris, M. D., "Factorial Sampling Plans for Preliminary Computational Experiments," *Technometrics*, Vol. 33, No. 2, 1991, pp. 161–174.
doi:10.2307/1269043

E. Livne
Associate Editor