JOURNAL OF PROPULSION AND POWER
Vol. 25, No. 5, September–October 2009

Comparison of Several Optimization Strategies for Robust Turbine Blade Design

Andy J. Keane*
University of Southampton, Southampton, England SO17 1BJ, United Kingdom

DOI: 10.2514/1.38673

This paper addresses the problem of turbine blade shape optimization in the presence of geometric uncertainties. Several strategies are tested and compared on a two-dimensional compressor blade optimization process for which performance is assessed using a commercial Reynolds-averaged Navier–Stokes computational fluid dynamics code. In each case, a range of shape errors are considered that attempt to simulate foreign object damage, erosion damage, and manufacturing errors. These lead to stochastic performance measures that, in turn, are considered in a multi-objective optimization framework. Because of the long run times associated with Reynolds-averaged Navier–Stokes codes, use is also made of surrogate or response-surface-based optimization methods to speed up the search processes. The paper shows that a range of techniques can be used to tackle this problem, but that no one method is clearly best overall. The practitioner is therefore cautioned against favoring a single approach for such design problems. Further research may help clarify these issues.

I. Introduction

THE use of optimization methods to design aerodynamic sections is now completely routine in many companies and research institutes. Typically a computer aided design (or other geometry design) system, meshing tool, computational fluid dynamics (CFD) solver, and postprocessor are used, coupled to a range of different optimization tools to give a design search and optimization capability. More recently, attention has turned to the issue of design robustness, because designs optimized purely for nominal performance may suffer from significantly degraded performance if they are subject to slight errors in manufacture, wear or damage in operation, or operation in conditions different to those at which they were optimized. This leads naturally to a multi-objective problem for which designers seek to improve the mean or nominal performance of a design while guaranteeing that falloff away from nominal conditions is strictly controlled. At the same time, the significant run times that can be encountered in using high-fidelity Reynolds-averaged Navier–Stokes (RANS) CFD codes for design optimization have been addressed recently by the use of surrogate or metamodel schemes to construct so called "response surfaces" that can help speed up search. In this paper we focus on both sets of approaches using industrial CFD tools and a large parallel computing cluster running the 64-bit Windows Server 2008 compute cluster edition operating system. The focus of this paper is on the two-dimensional optimization of a gas-turbine compressor blade section subject to damage in service and uncertainty in manufacture.

It is now commonplace to use design optimization methods to change the shape of aerodynamic sections when designing new gas turbines; see Keane and Nair [1] for an introduction to this approach. More recently, attention has focused on both robust multi-objective design (see for, example, Welch et al. [2] and Kumar et al. [3]) and surrogate-assisted design optimization (see, for example, Jones et al. [4] and Keane [5]). Approaches combining both these methods have also appeared (Keane [6] and Emmerich et al. [7]). Here we focus on the use of a series of shape-manipulation tools to explore the effects of foreign object damage, erosion, and uncertainty in manufacture while optimizing mean design performance and variance in a multi-objective framework. We compare and contrast four approaches. First, we simply optimize a baseline geometry for nominal performance using direct [8] and surrogate-based search [4]. Then, we employ a noisy phenotype [9] approach to allow for possible sensitivity in the mean performance but do not explicitly seek a tradeoff between mean performance and robustness. Third, we use a multi-objective evolutionary search [10] to deliberately construct a trade surface (Pareto front) adopting both direct and surrogate-assisted approaches. Finally, we use a surrogate-based multi-objective expected improvement formulation [6] to construct the tradeoff using single-objective search methods. Together these schemes represent the current state of the art of industrial practice in this area; by applying them with a set of powerful industrial analysis codes, it is possible to gain some insight into which approaches show the most promise for use in practical design work. It is of course true that more sophisticated direct stochastic solver methods are being researched in universities but these methods are, as yet, not widely adopted.

II. Geometry Modification

To carry out any form of automated aerodynamic section optimization, some form of geometry modification is required. Here we use a modified form of the parametric design and rapid meshing (PADRAM) code (Shahpar and Lapworth [11]) that has been described elsewhere (Kumar et al. [3]). In this process, a baseline airfoil section shape is altered by the addition of a series of Hicks–Henne functions in various ways (Hicks and Henne [12]). In the present approach, this scheme is used to make four sets of changes to the baseline airfoil that allow for overall design improvement and the modelling of manufacturing uncertainty, foreign object damage, and flank erosion. Figures 1a–1f illustrate the baseline airfoil section and each of these modification processes in turn. Note that all the changes are controlled such that the trailing-edge form and angle are kept fixed as would be needed to ensure unchanged inflow angles for the next stage in the compressor. Here, 10 Hicks–Henne functions are used for the design shape changes, distributed around the section, and this permits very significant alterations in geometry. In all the tables that follow, the normalized amplitude changes of these design variables are provided. These are defined in such a way that a value of 0.5 generates the base geometry, with values less than this removing material from the section and those greater adding to it; full details may be found in Kumar [13]. The other three sets of changes work in the same way but make much smaller shifts:

1) The manufacturing error changes affect the entire section (again using 10 Hicks–Henne functions distributed around the section, but

Received 2 June 2008; revision received 8 July 2009; accepted for publication 8 July 2009.

*Professor of Computational Engineering, School of Engineering Sciences, Highfield; ajk@soton.ac.uk.

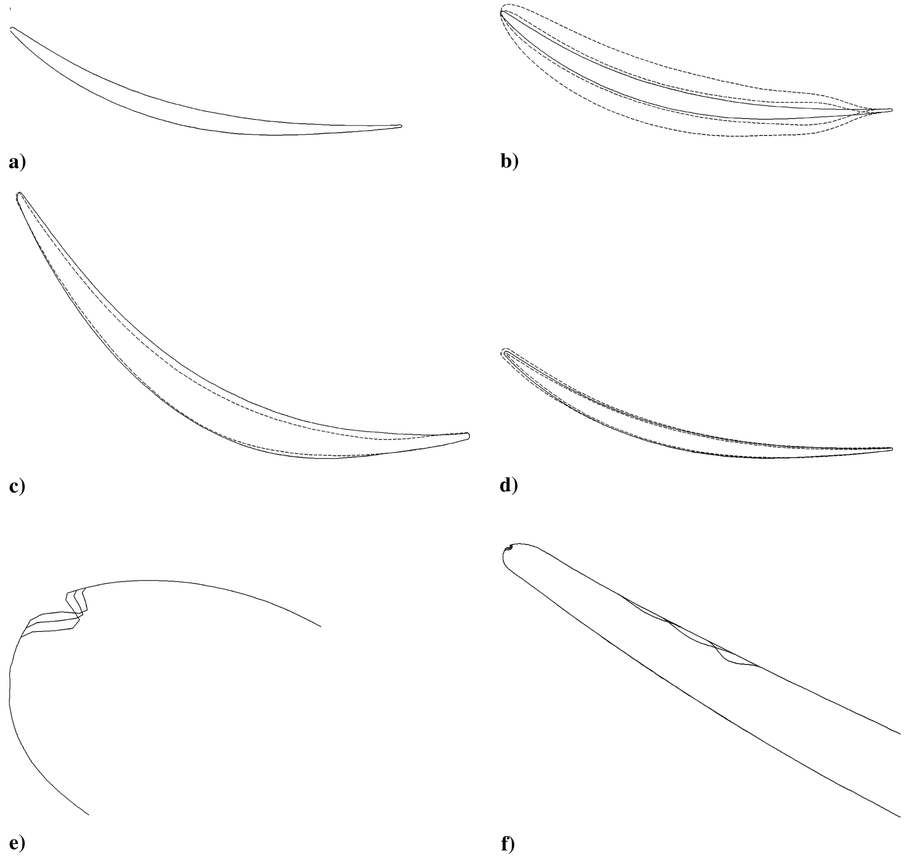

Fig. 1 Shown are the following: a) baseline airfoil section, b) overall design variation maximum possible changes, c) typical design change (shown distorted for clarity), d) maximum range of changes aiming to model uncertainty in manufacturing processes, e) leading-edge damage models, and f) flank erosion models.

now much smaller in scale) paying particular attention to the leading edge.

2) The foreign object damage changes are localized to the leading edge using a single Hicks–Henne function that can only remove material but can vary in position and shape as well as depth.

3) The erosion changes are localized to the flank, again using a single Hicks–Henne function that can again vary in position and shape as well as depth and can only remove material; it is applied where typical erosion events occur.

The modelling of the damage and erosion events is clearly far from accurate because the Hicks–Henne functions give smooth and continuous pockets whereas, in reality, these are typically irregular and sharp edged. However, as used here the pockets still act to trip and disturb the flow in a way that is not unrealistic. Note that the flank erosion pockets are allowed to extend over a greater area of the blade than the damage pockets at the tip.

III. Analysis

The PADRAM geometry modification process also permits the automated adaptation of high-quality "OCH" meshes around the airfoil section in a setting appropriate for application to gas-turbine blade design (here, OCH refers to the shapes of the various mesh blocks, which can be ringlike, cup shaped, or with legs). Figure 2 shows a typical mesh and the way that this mesh is distorted around the localized erosion geometries introduced when studying leading-edge damage or flank erosion. Here the O mesh is four cells deep and slightly over 27,000 cells are used in total. The topology of this mesh is unchanged throughout the study presented here so that any effects caused by remeshing can be avoided. If geometry change results in overly large distortions to the mesh, the relevant run is aborted and appropriate action is taken during any subsequent optimization or statistical processes.

Having set up the mesh, a full steady-state RANS solve can be carried out using the parallel version of the Hydra CFD code in around 40 min [14]. The section is run in isolation with boundary conditions with an inlet temperature of 290 K, an inlet total pressure of 63,400 Pa, a whirl angle of −37.3 deg, and an outlet static pressure of 52,000 Pa. Typical rms flow residuals are 4.7E-9 and rms turbulence residuals are 2.4E-15. Here we use pressure losses as the main design goal. The pressure losses are all normalized by the chosen baseline blade to preserve confidentiality; it therefore has a normalized loss coefficient of 1.0 (see column two of Table 1). Full validation studies of this meshing and solution process are presented by Kumar [13], who showed that the results achieved are reasonable for the kind of statistical work being presented here. Note that, by using this mesh, it is accepted that accurate modelling of the flow in the damage pockets cannot be achieved with this number of cells; in this study, the intention is that these regions should act as trips to the flow as rough-edge pockets would in practice. Clearly detailed flow analysis around an eroded section that attempted to capture all of the flow physics would require a significantly increased mesh count with consequent increases in run time.

If this baseline blade is then made subject to uncertainties in leading-edge damage, flank erosion, and manufacturing, it is possible to estimate the mean pressure loss and standard deviation in this loss using standard Monte Carlo approaches. To do this, 100 independent perturbations are made to each of the three sets of Hicks–Henne functions defining these three sets of uncertainties using a standard LPτ space-filling design of experiment (Statnikov and Matusov [15]) and the averages taken. The perturbations made are significantly beyond the extremes in tolerance band found in practice from manufacture and in service, but they provide a good measure of the resilience of a section design; such variations would be unlikely in practice. This calculation, of course, requires 100 times as much processing effort as finding the nominal performance, but it provides a datum for comparison; here the normalized mean loss

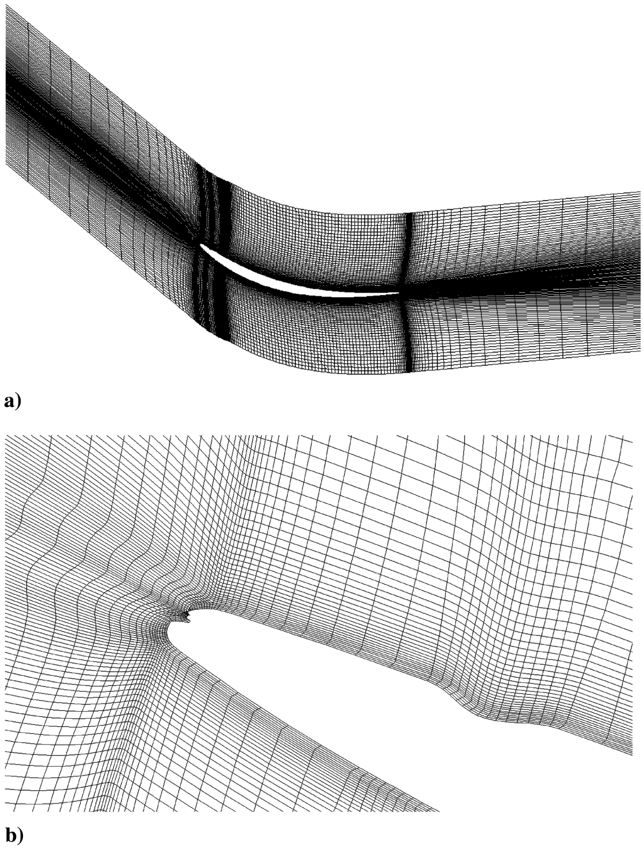

Fig. 2 PADRAM meshes: a) overall, and b) in way of leading-edge and flank erosion.

coefficient is 1.190 with a standard deviation of 0.133 (i.e., the mean loss is 19% worse than the deterministic calculation on the nominal baseline shape; see again column two of Table 1). Figure 3 shows the convergence history of this calculation; note that, because larger values of the perturbations almost inevitably lead to greater losses, the mean and standard deviation both rise until the 16-dimensional hypercube being sampled is fully explored by the pseudorandom number sequence in use, here after around 50 calculations; thereafter, the convergence is essentially unbiased. Unfortunately the use of such a large sample size in design searches cannot generally be justified. Consequently, 15 samples are used for most of the searches that follow and then the resulting geometries are checked with 100 samples to assess the likely true mean and standard deviations.

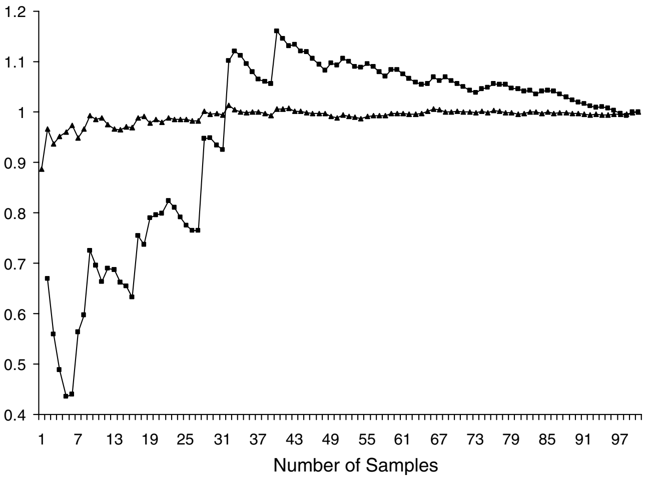

Fig. 3 Convergence of averaging process used in assessing uncertainties in shape (triangles show normalized mean and squares show normalized standard deviation).

IV. Nominal Performance Optimization

Having set up an automated analysis capability with associated geometry and mesh modification, it is then immediately possible to carry out an optimization against any given measurement of nominal performance. Here the converged Hydra CFD solution is used to calculate the pressure loss coefficient across the blade geometry, combined with the use of 10 Hicks–Henne functions to control the shape change. The process is then constrained so that designs that do not converge adequately within 500 CFD iterations are rejected (if rms flow residuals are worse than 5E-8 or rms turbulence residuals are worse than 5E-13). Because the mesh topology is fixed, the number of rejected designs is quite low, typically less than 2%, so that significant impact on the search process is avoided. The repeated application of a genetic algorithm (GA) search process (from the Options toolkit [16]) using a population of 50 and 15 generations allows designs with nominal normalized pressure losses in the range of 0.80–0.81 to be produced, an improvement of some 20% over the baseline design; see columns 3–5 of Table 1 (three runs with different random seeds are shown). Gains of this magnitude are typical of what is achievable when well-designed, but general-purpose sections are used as the starting point for this kind of optimization. Such an improvement is not, however, achieved without a negative aspect. If the nominal performance optimized blades are subject to leading-edge damage, flank erosion, and manufacturing uncertainty, it is again possible to estimate their mean pressure losses and the standard deviations in these losses using Monte Carlo approaches. As can be

Table 1 Comparisons of various blade designs, nominal single-objective optimizations

Quantity	Baseline design	Nominal GA direct optimized design			Nominal expected improvement optimized design		
Shape function 1	0.5	0.4063	0.4229	0.4125	0.4307	0.4263	0.4277
Shape function 2	0.5	0.4178	0.4342	0.4280	0.4279	0.4135	0.4164
Shape function 3	0.5	0.4069	0.4196	0.4022	0.4004	0.4041	0.4075
Shape function 4	0.5	0.4229	0.4083	0.4524	0.4001	0.4003	0.4166
Shape function 5	0.5	0.4060	0.4199	0.4164	0.4004	0.4170	0.4004
Shape function 6	0.5	0.5009	0.5089	0.5087	0.5049	0.4997	0.5015
Shape function 7	0.5	0.5496	0.5493	0.5425	0.5323	0.5278	0.5247
Shape function 8	0.5	0.4715	0.4589	0.4425	0.4511	0.4450	0.4531
Shape function 9	0.5	0.5478	0.5359	0.5457	0.5245	0.5235	0.5175
Shape function 10	0.5	0.5311	0.5210	0.5139	0.5205	0.5121	0.5104
Nominal loss	1.0000	0.8020	0.8031	0.8084	0.8171	0.8218	0.8274
Mean loss$_{100}$	1.1892	1.0097	1.0099	1.0055	1.0104	1.0277	1.0284
Std.dev.$_{100}$	0.1330	0.1542	0.1456	0.1461	0.1267	0.1416	0.1369
Average Mean loss$_{100}$	——		1.0084			1.0222	
Average std.dev.$_{100}$	——		0.1487			0.1351	
Total function calls per search	1		750			310	

seen from Table 1, the mean losses improve by 18% whereas, at the same time, the standard deviation is typically 12% greater. This is perhaps as might be expected when no account of the robustness of the design is made when conducting the design space searches. The lack of explicit control of sensitivity to manufacturing and in-service erosion is a major concern in carrying out deterministic design optimization and provides the driving motivation for the present study.

V. Response-Surface-Based Nominal Performance Optimization

Because optimization in which the analysis codes in use are directly coupled to the optimizer is a time-consuming process (direct optimization), it is now normal practice in most aerospace companies to apply some kind of surrogate-based response surface method (RSM) to help speed up design space searches. We follow this approach next and apply an "expected improvement" search using a Krig-based surrogate model following the logic of Jones et al. [4]. In this approach, a standard LPτ space-filling design of experiments (DOE) of 150 sample designs is analyzed first, followed by a series of updates that are used to enrich and refine the available data set. These update points are positioned as indicated by the expectation of improvement that can be calculated given the underlying assumption of Gaussian standard processes inherent in Kriging. The expectation explicitly balances the likelihood of finding an improved design with that of improving the quality of the response surface being constructed and, thus, the reliability of the search, that is, it balances exploitation and exploration. This is in contrast to simply placing new update points at those locations where the Krig predicts the best nominal designs to lie; such a scheme might miss good regions of the design space if the initial DOE had failed to sample them. Here we also take particular care to carefully adjust the hyper-parameters of the Krig before searching for batches of updates located at the extrema of the improvement metric. In total, 16 such batches each of 10 designs are used, leading to a total evaluation count of 310, see columns 6–8 of Table 1. In these cases, the search has produced slightly worse nominal losses than for the direct search, which also have very slightly worse means when evaluated over 100 design perturbations, though with less than half the computational effort. They still improve the mean performance by 17% when compared with the base design. Additionally, the standard deviations are marginally improved, in one case better even than for the baseline. Given that no direct control is placed on variance, this is not surprising.

Having illustrated what may be obtained by focusing exclusively on nominal design performance, we turn next to methods that seek to take explicit account of uncertainties, albeit still in a single-objective setting.

VI. Noisy Phenotype Optimization

To help overcome the tendency of nominal performance optimization to produce overly sensitive designs, a number of researchers have suggested that some form of perturbation scheme be used during search. These typically work by substituting the nominal performance of the design during optimization by some other value stemming from some kind of uncertainty-based geometry. These are generally classed as "noisy phenotype" models in the GA literature [9] and can involve various numbers of random samples being taken, with mean or worst-case values being used in lieu of the nominal performance metrics. Here we follow two approaches. In the first, we carry out two analyses, one on the nominal geometry and the second on a randomly chosen geometry, lying within the space of permissible uncertainties. We then return the worst of these two values as the calculated loss metric. In the second, we sample the uncertainty space with 15 sets of uncertainty parameters chosen following an LPτ design of experiments plan and return the mean loss. The first scheme is clearly much faster than the second but does involve using a nonstationary objective function in the search, because repeated analyses of the same nominal geometry will return different objective function values, and this is known to slow down the convergence of search methods. Moreover, by working with only a single perturbation to the design, the metric being fed to the optimizer is often not a very accurate representation of the performance when averaged over 100 samples.

In both the cases tested here, we apply a direct GA design space search from the Options toolkit to the problem for 15 generations, each of 50 members. These searches give rise to the results in columns 3–5 and 6–8 of Table 2 (again each search is run 3 times, with different random seeds, to illustrate the variance that can arise). As can be seen from the table, the loss values found during the searches do not agree very well with those for the 100-sample Monte Carlo calculations, particularly for the two-sample approach. Moreover, the designs are little better than those coming from the simple direct optimizations, in that both the mean and standard deviations taken over 100 samples are broadly similar using all four approaches (the standard deviations for the two-sample approach are marginally better, their mean performance 3% worse). That for the 15-sample noisy phenotype approach also represents an extremely time-consuming search calculation; it can therefore not be recommended as a useful approach in design work unless the standard deviation results are also explicitly used in the search. As will be seen next, the standard deviations are better controlled when this quantity is used directly during optimization. Even so, the two-sample approach is a simple and relatively efficient way of ensuring that some degree of robustness in the results is at least reflected in the search, a point that has been demonstrated before [17]. (Note that it is not possible to use RSM-based

Table 2 Comparisons of various blade designs, noisy phenotype single-objective optimizations

Quantity	Baseline design	Noisy phenotype optimized design (two samples)			Noisy phenotype optimized design (15 samples)		
Shape function 1	0.5	0.4287	0.4106	0.4318	0.4017	0.4063	0.4007
Shape function 2	0.5	0.4076	0.4454	0.4008	0.4118	0.4120	0.4271
Shape function 3	0.5	0.4767	0.4235	0.4740	0.4063	0.4069	0.4352
Shape function 4	0.5	0.4096	0.4735	0.5105	0.4193	0.4980	0.4417
Shape function 5	0.5	0.4420	0.4409	0.4561	0.4667	0.4143	0.4854
Shape function 6	0.5	0.5244	0.4971	0.5174	0.5121	0.5133	0.5070
Shape function 7	0.5	0.5081	0.4983	0.5490	0.5405	0.5406	0.5317
Shape function 8	0.5	0.4025	0.4672	0.4883	0.4762	0.4481	0.4647
Shape function 9	0.5	0.5380	0.5315	0.5359	0.5410	0.5082	0.5283
Shape function 10	0.5	0.4813	0.5078	0.5080	0.5209	0.4985	0.4722
Nominal loss	1.0000	0.8747	0.9025	0.8222	1.0000	0.8157	0.8317
Mean loss$_{2,15}$	——	0.9226	0.9024	0.8735	0.9526	0.9631	0.9700
Std.dev.$_{2,15}$	——	0.0339	0.0184	0.0364	0.0908	0.0886	0.0884
Mean loss$_{100}$	1.1892	1.0574	1.0678	1.0291	1.0051	1.0091	1.0176
Std.dev.$_{100}$	0.1330	0.1180	0.1263	0.1397	0.1436	0.1385	0.1339
Average mean loss$_{100}$	——		1.0515			1.0106	
Average std.dev.$_{100}$	——		0.1280			0.1387	
Total function calls per search	1		1500			11250	

approaches to speed up the two-sample approach because at each step there are insufficient results to produce RSMs and the results are too noisy to span the whole space with a single RSM. The 15-sample approach can be tackled with RSMs as will be seen later.)

VII. Tradeoff Analysis

Most practical design problems have more than one goal that the designer tries to improve. In aerospace design, it is common to aim for lightweight, low-cost, robust, high-performance systems. These aims are often in tension with each other, and so compromise solutions have to be sought. Such compromises inevitably involve deciding on some form of weighting between the goals. However, before this stage is reached, it is possible to study design problems from the perspective of Pareto sets. A Pareto set of designs is one for which the members are all optimal in some sense, but for which the relative weighting between the competing goals is yet to be finally fixed (see, for example, Fonseca and Fleming [18]). More formally, a Pareto set of designs contains systems that are sufficiently optimized that, to improve the performance of any set member in any one goal function, its performance in at least one of the other functions must be made worse. Moreover, the designs in the set are said to be "non-dominated," in that no other set member exceeds a given design's performance in *all* goals.

Currently, there appear to be two popular ways of constructing Pareto sets; see, for example, Keane and Nair [1]. First, and most simply, one chooses a (possibly nonlinear) weighting function to combine all the goals in the problem of interest into a single quantity and carries out a single-objective optimization. The weighting function is then changed and the process repeated. By slowly working through a range of weightings, it is possible to build up a Pareto set of designs. This approach allows the full range of single-objective search methods to be applied, including the use of DOE and RSM schemes to speed up the search as per the previous section. It does, however, suffer from a major drawback: it is often not clear what weighting function to use and how to alter it so as to be able to reach all parts of the potential design space (and, thus, to have a wide-ranging Pareto set). The nonlinear nature of most design problems will make it very difficult to ensure that the designs achieved are reasonably evenly spaced out through the design space. Moreover, if the Pareto front is concave, then highly nonlinear weighting functions must be used if all regions of the front are to be reached.

To address these limitations, designers have resorted to a second way of constructing Pareto sets via the use of population-based search schemes. In such schemes, a set of designs is worked on concurrently and evolved toward the final Pareto set in one process. In doing this, designs are compared with each other and progressed if they are of high quality *and* if they are widely spaced apart from other competing designs. Moreover, such schemes usually avoid the need for an explicit weighting function to combine the goals being studied. One of the most well known of these schemes is the nondominated sorting genetic algorithm, version 2 (NSGA-ii) method introduced by Deb et al. [10]. In this approach, a GA is used to carry out the search but the goal function used to drive the genetic process is based on the relative ranking and spacing of the designs in the set rather than their combined weighted performance. More specifically, at each generation, all the designs are compared and the nondominated designs are set to one side. These are assigned rank 1. The remaining designs are compared, and those that now dominate are assigned rank 2, and so on. Thus, the whole population is sorted into rank order based on dominance. This sorting into rank order dominance can be carried out irrespective of the relative importance of the objectives being dealt with or the relative magnitudes and scaling of these quantities.

Having sorted the population of designs into ranks, they are next rewarded or penalized depending on how close they are to each other in the goal space. To do this, the Euclidean distance to the nearest rank neighbor of each design is calculated and then normalized by the closest distance in the current rank set. Each design is then increasingly penalized the closer it is to a neighbor in a simple linear fashion. An additional reward is provided to those designs at the ends of the front. This process induces the search algorithm to explore the whole design space, but does require that the competing objectives be suitably scaled, an important issue that arises in many aspects of dealing with multi-objective approaches to design. When combined with the traditional GA operators of selection, crossover, and mutation, the NSGA-ii scheme is remarkably successful in evolving high-quality Pareto sets. As originally described, however, no means were provided for mitigating run time issues arising from using expensive computer simulations in assessing competing designs. Moreover, the genetic operators used were somewhat simplistic; more sophisticated GA search methods are now commonly invoked, as here.

To overcome the problem of long execution times, a number of researchers have used RSM approaches, including Kriging, within Pareto front frameworks (Wilson et al. [19] and Knowles and Hughes [20]). It is also possible to combine tools such as NSGA-ii with Kriging (Voutchkov and Keane [21]). In such schemes, an initial DOE is carried out and RSMs built as per the single-objective case, but now there is one RSM for each goal function. In the NSGA-ii approach, the search is simply applied to the resulting RSMs and used to produce a Pareto set of designs. These designs are then used to form an update set and, after running full computations, the RSMs are refined and the approach continued. Although sometimes quite successful, this approach suffers from an inability to explicitly balance exploration and exploitation in the RSM construction in just the same way as when using such models in a single-objective search, greedily seeking for the best designs, although the crowding or niching measures normally used help mitigate these problems to some extent. These two approaches are used next to construct robust blade sections.

VIII. Direct NSGA-ii Optimization

Columns 3–5 of Tables 3 and 4 show six of the results of using a direct NSGA-ii search on the problem being studied here, using 15 generations and a population of 50 members. Those in Table 3 are taken from the low-loss end of three Pareto fronts produced using differing seeds in the searches, whereas those in Table 4 come from around the midpoints of the fronts, offering a balance between the mean loss and standard deviation in loss; see for example Fig. 4 (the two ends of the front in the figure represent designs for which only the mean loss or only standard deviation of the loss are optimized). These designs have been found using the mean and standard deviation of performance estimated using 15 samples, again chosen following an LPτ design of experiments plan, that is, with the same computational effort as for the 15-sample noisy phenotype design. Again, these results do not align closely with the full 100-sample calculation but they are sufficiently representative to allow the multi-objective search to deliver useful results: the low-loss designs are comparable to those coming from the direct search of the 15-sample noisy phenotype approach. The balanced designs are significantly more robust than any of those seen so far in this study; they have a mean performance some 5% worse than the low-loss optimized designs but 28% better standard deviations than any of the single-objective designs or the baseline.

IX. Surrogate-Assisted NSGA-ii Optimization

Next, in an effort to reduce the run times associated with direct search on this problem, a Kriging-based surrogate-assisted NSGA-ii approach is applied; see columns 6–8 of Tables 3 and 4 and Fig. 5. These results are almost as good as those for the direct NSGA-ii approach and have been achieved using an initial sample of 150 designs to construct the baseline Krig models followed by seven updates each of 25 new points, but still with the 15-point uncertainty sampling plan, that is, using around 43% the effort of the direct search. In fact, it is clear from Fig. 5 that the Krig-based Pareto front compares well with the final version produced from the direct search after just three updates, which amounts to only 30% of the computational effort needed. One drawback of this approach is that after seven updates further improvements become difficult to find as the

Table 3 Comparisons of various blade designs, taken from Pareto fronts; minimum loss designs

Quantity	Baseline design	NSGA-ii direct optimized design			NSGA-ii surrogate optimized design			Expected improvement optimized design		
Shape function 1	0.5	0.4015	0.4062	0.4015	0.4015	0.4019	0.4015	0.4301	0.4313	0.4241
Shape function 2	0.5	0.4135	0.4119	0.4207	0.4529	0.4162	0.4183	0.4121	0.4475	0.4451
Shape function 3	0.5	0.4257	0.4006	0.4381	0.4006	0.4381	0.4381	0.4590	0.4008	0.4482
Shape function 4	0.5	0.4204	0.4158	0.4204	0.4029	0.4204	0.4204	0.5083	0.5410	0.4516
Shape function 5	0.5	0.5047	0.4518	0.4395	0.4680	0.4678	0.4747	0.4788	0.4010	0.4468
Shape function 6	0.5	0.5120	0.5030	0.5187	0.5263	0.5132	0.5129	0.5334	0.5071	0.5141
Shape function 7	0.5	0.5406	0.5406	0.5191	0.5014	0.5193	0.5187	0.5347	0.5365	0.5109
Shape function 8	0.5	0.4637	0.4797	0.4641	0.4743	0.4590	0.4637	0.4612	0.4620	0.4540
Shape function 9	0.5	0.5418	0.5021	0.5293	0.5490	0.5495	0.5410	0.5462	0.5499	0.5008
Shape function 10	0.5	0.4987	0.4984	0.4893	0.4918	0.4785	0.5373	0.5266	0.4787	0.4544
Nominal loss	1.0000	0.8118	0.8161	0.8385	0.8786	0.8410	0.8435	0.8534	0.8295	0.8684
Mean loss$_{15}$	——	0.9560	0.9676	0.9623	0.9779	0.9656	0.9686	1.0111	0.9954	0.9956
Std.dev.$_{15}$	——	0.0910	0.0934	0.0798	0.0634	0.0812	0.0814	0.0798	0.0833	0.0740
Mean loss$_{100}$	1.1892	1.0082	1.0185	1.0059	1.0183	1.0109	1.0157	1.0353	1.0183	1.0393
Std.dev.$_{100}$	0.1330	0.1437	0.1455	0.1232	0.1141	0.1217	0.1231	0.1205	0.1253	0.1274
Average mean loss$_{100}$	——		1.0109			1.0149			1.0310	
Average std.dev.$_{100}$	——		0.1374			0.1197			0.1244	
Total function calls	1		11250			4875			4500	

Table 4 Comparisons of various blade designs, taken from Pareto fronts; balanced designs

Quantity	Baseline design	NSGA-ii direct optimized design			NSGA-ii surrogate optimized design			Expected improvement optimized design		
Shape function 1	0.5	0.4138	0.4252	0.4015	0.4195	0.4013	0.4302	0.4142	0.4300	0.4289
Shape function 2	0.5	0.4225	0.4974	0.4535	0.5025	0.4114	0.5017	0.5401	0.4373	0.4416
Shape function 3	0.5	0.4067	0.4066	0.4207	0.4207	0.4381	0.4207	0.4344	0.4429	0.5182
Shape function 4	0.5	0.4013	0.5065	0.4204	0.4144	0.4206	0.4074	0.4318	0.4679	0.4042
Shape function 5	0.5	0.4373	0.4173	0.4984	0.4743	0.4360	0.4675	0.4725	0.4855	0.4718
Shape function 6	0.5	0.5265	0.5383	0.5339	0.5280	0.5370	0.5333	0.5330	0.5366	0.5201
Shape function 7	0.5	0.4808	0.4752	0.4763	0.4805	0.4761	0.4922	0.5279	0.5112	0.5070
Shape function 8	0.5	0.4637	0.4633	0.4641	0.4696	0.4424	0.4719	0.4497	0.4529	0.4560
Shape function 9	0.5	0.5410	0.5176	0.5224	0.5182	0.5365	0.5260	0.4570	0.5087	0.4619
Shape function 10	0.5	0.4984	0.4998	0.4770	0.4983	0.5163	0.4892	0.5010	0.4144	0.4201
Nominal loss	1.0000	0.9189	0.9690	0.9455	0.9497	0.9369	0.9265	0.9025	0.9047	0.9072
Mean loss$_{15}$	1.1471	1.0028	1.0453	1.0121	1.0314	1.0144	1.0231	1.0329	1.0128	1.0208
Std.dev.$_{15}$	0.0682	0.0505	0.0423	0.0456	0.0471	0.0532	0.0551	0.0544	0.0490	0.0531
Mean loss$_{100}$	1.1892	1.0412	1.0769	1.0529	1.0634	1.0507	1.0580	1.0808	1.0677	1.0764
Std.dev.$_{100}$	0.1330	0.1047	0.0956	0.0982	0.0965	0.0963	0.1066	0.1183	0.1151	0.1196
Average mean loss$_{100}$	——		1.0570			1.0573			1.0750	
Average std.dev.$_{100}$	——		0.0995			0.0998			0.1177	
Total function calls	1		11250			4875			4500	

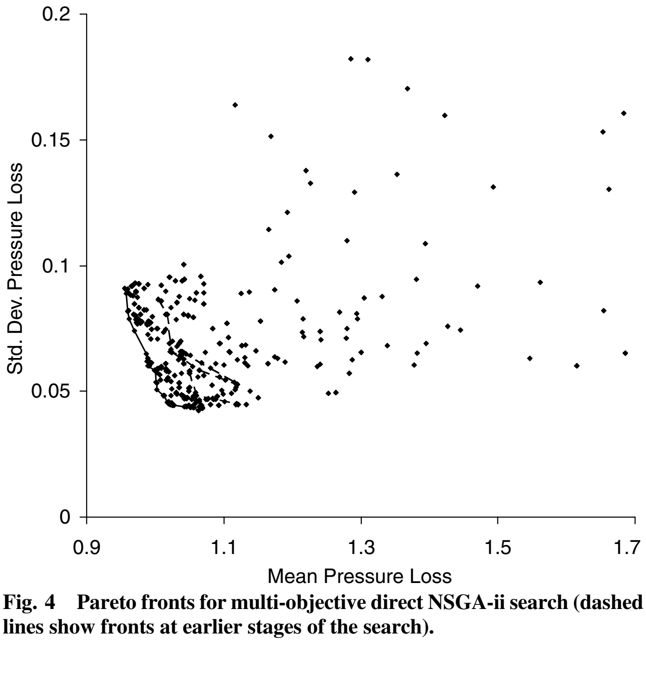

Fig. 4 Pareto fronts for multi-objective direct NSGA-ii search (dashed lines show fronts at earlier stages of the search).

Krig models are not as good at uncovering novel new behavior as the direct NSGA-ii search, because they are always based on points from previous updates. The process, therefore, fails to fully explore the ends of the Pareto front.

X. Expected Improvement Optimization

Lastly, an expected improvement formulation [6] is used with Krig-based surrogates of the mean and standard deviation produced from the same 15-point uncertainty sampling plan used with the NSGA-ii schemes. In this method, however, the statistical model available in the Krig models is used to calculate the expectation of finding an improvement over the calculated Pareto front. Thus, after an initial sample of 150 designs has been studied, 15 such promising locations are chosen by applying an optimizer to the predicted expectation of improvement, and then these points are used to update the Krig models with 10 such update sequences being used in total, leading to a total study of 300 different designs, each sampled 15 times. Because the expectation again balances exploitation and exploration, it is less likely to stall than the RSM-based NSGA-ii while still profiting from the efficiency of the surrogate modelling. The results from this search are shown in columns 9–11 of Tables 3 and 4 and in Fig. 6. As can be seen from the figure, the resulting search is competitive with the direct RSM-based approach in terms of mean loss designs but now includes a range of low standard deviation results not found by the other multi-objective searches. It is not so

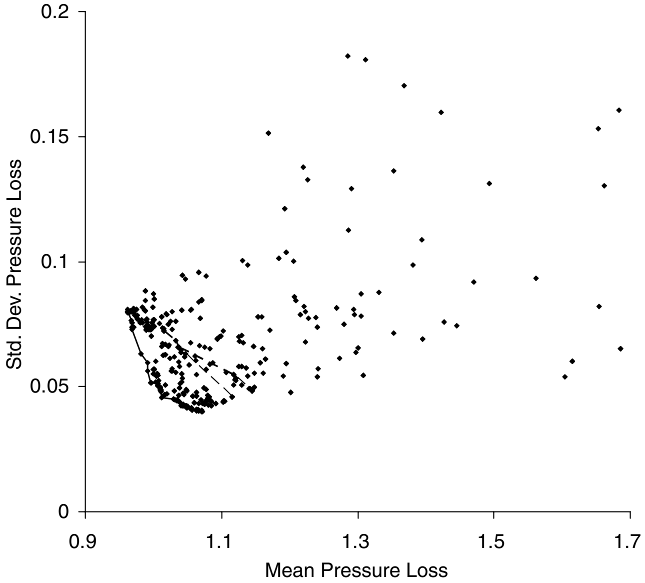

Fig. 5 Pareto fronts for multi-objective surrogate-assisted NSGA-ii search (dashed lines show fronts at earlier stages of the search).

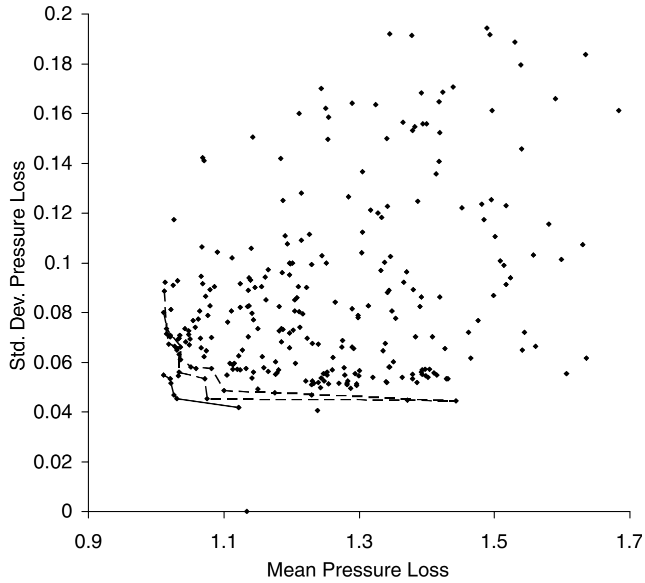

Fig. 6 Pareto fronts for multi-objective expected improvement search (dashed lines show fronts at earlier stages of the search).

good at locating the very lowest loss designs, however, being 1.5–2% worse in this respect. It is noticeable, however, that the Pareto front is less converged in this case and so longer runs with more updates may have improved on this aspect. These results indicate that it is carrying out a more careful exploration of the design space, although slightly more slowly than the RSM-based NASG2 run.

Finally, Fig. 7 shows the geometries of the nine balanced designs detailed in Table 4 alongside the base design. As can be seen, all the robust designs show similar changes to the pressure face that improve the overall performance, with noticeable variations in the last third of the section, whereas the first two thirds are all broadly similar.

XI. Variations in Calculated Results

In all the searches reported here, some form of random number or pseudorandom number sequence has been used. Sometimes these are required because searches such as the GA rely on random operations to work, whereas in space-filling designs a pseudorandom sequence is used to sample the design space. It is therefore the case that all the results reported would vary with different sequences of random numbers and so, to draw more definite conclusions, some more thorough investigation of averaging than just three independent runs would be needed; unfortunately, when working with resource intensive RANS codes, as here, this is not possible even given the extensive computing power available in the cluster being used. The reader is therefore cautioned that, without much more extensive and long running testing, no firm guarantees can be offered for the relative rankings of the methods described; all the work presented here must be treated as indicative of typical behavior. It does, however, demonstrate the clear benefit of including variance measures in design optimization. Finally, it should be noted that it is current practice when designing blades for aeroengines to work mainly with three-dimensional CFD models; the use of two-dimensional approaches here has been adopted simply to limit the computational cost of the study.

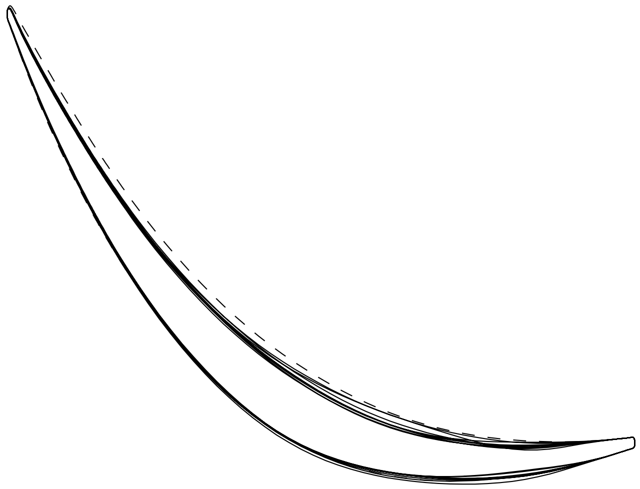

Fig. 7 Balanced designs from Table 4 showing variations in geometry, with base design shown dashed (shown distorted for clarity).

XII. Conclusions

In this paper the robustness of two-dimensional gas-turbine compressor blade sections is studied. The designs considered are subjected to variations that model uncertainty in manufacture and in-service shape degradation. The performance evaluation is carried out with an industrial Reynolds-averaged Navier–Stokes code. A range of different schemes are used to propagate uncertainty through the analysis process, and these are combined with state-of-the-art methods to limit run times during optimization. The results presented show that good-quality Krig RSMs are capable of supporting such optimizations and that noisy phenotype, reduced-size Monte Carlo sampling schemes can yield designs with improved mean performance. When linked to multi-objective optimization schemes an explicit balance between performance and robustness can also be achieved. Ultimately, a multi-objective RSM supported GA seems to offer the best balance between effort and the quality of the resulting designs.

Acknowledgment

The use of Rolls-Royce plc codes, their financial support, and the assistance of Shahrokh Shahpar and Apu Kumar in preparing the computational framework to support this research are all gratefully acknowledged.

References

[1] Keane, A. J., and Nair, P. B., *Computational Approaches for Aerospace Design*, Wiley, Chichester, England, U.K., 2005.

[2] Welch, W., Yu, T., Kang, S., and Sacks, J., "Computer Experiments for Quality Control by Parameter Design," *Journal of Quality Technology*, Vol. 22, 1990, pp. 15–22.

[3] Kumar, A., Keane, A. J., Nair, P. B., and Shahpar, S.,"Robust Design Method of Compressor Fan Blades Against Erosion," *Journal of Mechanical Design*, Vol. 128, 2006, pp. 864–873.
doi:10.1115/1.2202886

[4] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.
doi:10.1023/A:1008306431147

[5] Keane, A., "Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods," *Journal of Aircraft*, Vol. 40, 2003, pp. 741–750.
doi:10.2514/2.3153

[6] Keane, A., "Statistical Improvement Criteria for Use in Multiobjective Design Optimization," *AIAA Journal*, Vol. 44, 2006, pp. 879–891.
doi:10.2514/1.16875

[7] Emmerich, M. T. M., Giannakoglou, K. C., and Naujoks, B., "Single and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Transactions on Evolutionary Computation*, Vol. 10, 2006, pp. 421–439.
doi:10.1109/TEVC.2005.859463

[8] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, MA, 1989.

[9] Tsutsui, S., and Ghosh, A., "Genetic Algorithms with a Robust Solution Searching Scheme," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 3, 1997, pp. 201–208.
doi:10.1109/4235.661550

[10] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-ii," *Lecture Notes in Computer Science*, Vol. 1917, 2000, pp. 848–849.
doi:10.1007/3-540-45356-3_83

[11] Shahpar, S., and Lapworth, L., "Parametric Design and Rapid Meshing Systems for Turbomachinery Optimization," American Society of Mechanical Engineers Paper GT2003-38698, 2003.

[12] Hicks, R. M., and Henne, P., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, 1978, pp. 407–412.
doi:10.2514/3.58379

[13] Kumar, A., "Robust Design Methodologies: Application to Compressor Blades," Ph.D. Thesis, Univ. of Southampton, Southampton, England, U.K., 2006.

[14] Moinier, P., and Giles, M. B., "Preconditioned Euler and Navier–Stokes Calculations on Unstructured Grids," *Proceedings of the 6th ICFD Conference on Numerical Methods for Fluid Dynamics*, The Institute for Computational Fluid Dynamics, Oxford, 1999.

[15] Statnikov, R. B., and Matusov, J. B., *Multicriteria Optimization and Engineering*, Chapman and Hall, New York, 1995.

[16] Keane, A. J., *The Options Design Exploration System. Reference Manual and User Guide*, Univ. of Southampton, Southampton, England, U.K., 2003; also http://www.soton.ac.uk/~ajk/options.ps [retrieved 21 July 2009].

[17] Anthony, D. K., and Keane, A. J., "Robust-Optimal Design of a Lightweight Space Structure Using a Genetic Algorithm," *AIAA Journal*, Vol. 41, 2003, pp. 1601–1604.
doi:10.2514/2.2114

[18] Fonseca, C. M., and Fleming, P. J., "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, Vol. 3, 1995, pp. 1–16.
doi:10.1162/evco.1995.3.1.1

[19] Wilson, B., Cappelleri, D., Simpson, W., and Frecker, M., "Efficient Pareto Frontier Exploration Using Surrogate Approximations," *Optimization and Engineering*, Vol. 2, 2001, pp. 31–50.
doi:10.1023/A:1011818803494

[20] Knowles, J. D., and Hughes, E. J., "Multiobjective Optimization on a Budget of 250 Evaluations," *Lecture Notes in Computer Science*, Vol. 3410, 2005, pp. 176–190.
doi:10.1007/b106458

[21] Voutchkov, I. I., and Keane, A. J., "Multiobjective Optimization Using Surrogates," *Proceedings of the 7th International Conference on Adaptive Computing in Design and Manufacture*, Springer, Berlin, 2006, pp. 167–175.

F. Liu
Associate Editor