Applied Soft Computing 13 (2013) 1882–1891

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Differential evolution based soft optimization to attenuate vane–rotor shock interaction in high-pressure turbines

Michael M. Joly*, Tom Verstraete, Guillermo Paniagua

*von Karman Institute for Fluid Dynamics, Chaussée de Waterloo 72, B-1640 Rhode Saint Genèse, Belgium*

A R T I C L E I N F O

*Article history:*
Received 22 September 2010
Received in revised form
29 November 2012
Accepted 25 December 2012
Available online 11 January 2013

*Keywords:*
Design optimization
Evolutionary algorithm
Turbomachinery
Vane–rotor shock interaction

A B S T R A C T

This article presents a soft computing methodology to design turbomachinery components experiencing strong shock interactions. The study targets a reduction of unsteady phenomena using evolutionary optimization with robust, high fidelity, and low computational cost evaluations. A differential evolution (DE) algorithm is applied to optimize the transonic vane of a high-pressure turbine. The vane design candidates are examined by a cost-effective Reynolds-averaged Navier–Stokes (RANS) solver, computing the downstream pressure distortion and aerodynamic efficiency. A reduction up to 55% of the strength of the shock waves propagating downstream of the stand-alone vane was obtained. Subsequently to the vane optimization, unsteady computations of the vane–rotor interaction were performed using a non-linear harmonic (NLH) method. Attenuation above 60% of the unsteady forcing on the rotor (downstream of the optimal vane) was observed, with no stage-efficiency abatement. These results show the effectiveness of the proposed soft optimization to improve unsteady performance in modern turbomachinery exposed to strong shock interactions.



## 1. Introduction

In the development of any modern aeroengine, the most expensive component is the high-pressure turbine due to the harsh environment (high temperatures downstream of the combustor and mechanical solicitations). Higher loading per row allows to reduce the number of stages, limiting the weight of the machine. It contributes therefore to lower the fuel consumption of commercial aircrafts. However, an increase of load implies that the flow across the turbine passages is transonic, resulting in shock-wave interactions [1]. Denton et al. [2] describes the aerodynamics of the trailing-edge shock system within transonic turbine vanes. The vane shocks waves travel downstream, impacting periodically on the rotor blades (see Fig. 1, left). Giles [3] identified the sweeping of the direct shock from the crown of the rotor blade towards the leading edge, causing variations in the rotor lift of 40% of the mean level. The downstream rotor is therefore prone to suffer from high cycle mechanical and thermal fatigue.

Attempts to mitigate the unsteady vane–rotor shock interaction could be classified into improved designs through a better physical understanding [4–7], active control systems [8], and numerical optimizations with neural networks [9,10]. Unsteady turbine stage computations with high fidelity are however extremely expensive and their implementation together with optimization algorithms is limited to 2D profiles. The unsteady optimization of real 3D geometries remains unpractical with modern computers, even with the assistance of surrogate models. The present research proposes alternatively a soft computing methodology based on evolutionary optimization, and considers robust, accurate, and computationally affordable evaluations to redesign the vane, with the ultimate goal to limit the unsteady vane–rotor shock interaction.

It is proposed to modulate directly the pitch-wise static pressure distribution downstream of the stand-alone vane with Reynolds-averaged Navier–Stokes (RANS) computations, as suggested by Shelton et al. [11]. The aim is to attenuate the strength of the shock waves that propagates downstream of the vane. The soft optimization is expressed as a multi-objective problem. A differential evolution (DE) algorithm is used and assessed on mathematical test cases. For the vane design a parameterization of the two-dimensional section is developed with particular focus on the contraction channel. Another parameterization of the stacking line allows to introduce lean of the three-dimensional geometry. Each candidate is processed by an automatic structured mesh generator and evaluated by Navier–Stokes computations. In the result sections, the obtained optimal two- and three-dimensional vane geometries are presented. Their flow features are analyzed to understand how the optimized geometry reduces the downstream propagation of shock waves. Subsequently to the vane optimization, the abatement in the rotor forcing was quantified using an unsteady solver based on a non-linear harmonic (NLH) method [12].

* Corresponding author.

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.12.005

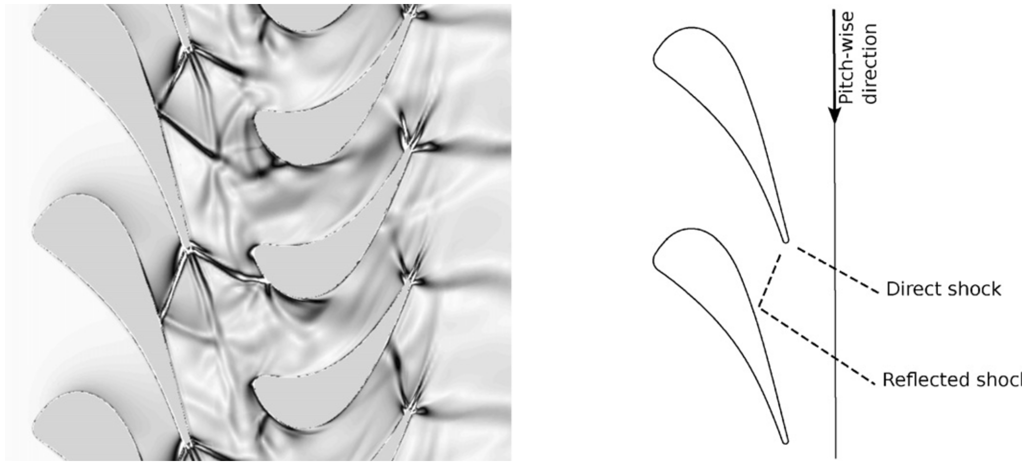


**Fig. 1.** (Left) Vane/rotor shock interaction (Paniagua et al. [7]). (Right) Plane of interest to assess vane downstream distortion.

## 2. Soft optimization methodology

### 2.1. A multi-objective optimization problem

The optimization of the stand-alone vane has two objectives. The first objective is to minimize the distortion of the pitch-wise static pressure at the vane outlet. Fig. 1, right, displays the location where the pressure is evaluated, 35% of the axial chord downstream of the vane trailing edge. The distortion downstream of the vane is assessed by the standard deviation along the pitch-wise direction, expressed by Eq. (1):

$$\sigma = \sqrt{\int_{y_0}^{y_0+\text{pitch}} \frac{(p(1.35 \times \lambda, y) - \overline{p})^2}{\text{pitch}}\, dy} \quad (1)$$

The second objective is to ensure high efficiency. Therefore the kinematic loss coefficient (Eq. (2)) ought to be minimized.

$$\xi = 1 - \eta = 1 - \frac{V_2^2}{V_{2is}^2} \quad (2)$$

### 2.2. Differential evolution

A multi-objective optimization is commonly described as follows with a set of objective functions $f_i(\vec{x})$ to be minimized.

$$\begin{array}{lll} \text{Minimize} & f_i(\vec{x}) & i = 1, \ldots, l \\ \text{Subject to} & g_j(\vec{x}) \leqslant 0 & j = 1, \ldots, m \\ & x_p^u(\vec{x}) \leqslant x_p \leqslant x_p^v(\vec{x}) & p = 1, \ldots, n \end{array}$$

A design vector $\vec{x}$ is defined by a set of parameters $x_p$; each one being bounded in a specific design range. The set of optimal parameters presenting minimal objective function values should also respect the conditions defined by the constraints $g_i(\vec{x})$.

Evolutionary algorithms (EAs) are well suited to multi-objective optimization problems [13]. They are based on Darwinian evolution whereby populations of individuals evolve over a search space and adapt to the environment by the use of different mechanisms such as mutation, crossover, and selection. Individuals with a higher fitness have more probability to survive and/or get reproduced. EAs are also capable to handle complex problems, involving features such as discontinuities, multi-modality, disjoint feasible spaces and noisy function evaluations.

In the present work the differential evolution (DE) method is used. DE is a relatively recent evolutionary method developed by Price and Storn [14]. Compared to genetic algorithms (GA) [15], the method does not require the transformation of continuous variables into binary strings. The method of Madavan [16] allows to extend the algorithm to multi-objective problems; it uses the non-dominated sorting and ranking selection scheme of Deb et al. [17]. In his paper, Madavan reported that the method is self-adaptive, elitist, and capable to maintain diversity in the Pareto set. The efficacy and capabilities of the algorithm were demonstrated with several complex test problems.

Two typical two-dimensional mathematical optimization problems are used to verify the correct implementation of the method for the current research. The first one is formulated as follows.

$$\begin{array}{ll} \text{Minimize} & f1(\vec{x}) = 5\sin(\pi x1) + \cos(\pi x2) \\ & f2(\vec{x}) = \sin(\pi x1)\cos(\pi x2) + x1x2 \\ \text{Subject to} & -1 \leqslant x1 \leqslant 1 \\ & -1 \leqslant x2 \leqslant 1 \end{array}$$

The constants used by the DE algorithm are $F = 0.3$ and $C = 0.8$. The optimization problem results into a discontinuous Pareto front expressed with Eqs. (3) and (4).

$$f_2 = -\frac{f_1+1}{5} + \frac{1}{\pi} a \sin\left(\frac{f_1+1}{5}\right) - 1 \quad (3)$$

$$f_2 = -f_1 - 5 - \frac{1}{2\pi} a \cos(f_1 + 5) \quad (4)$$

The second test problem is the ZDT3 function. It includes two objectives, expressed with Eqs. (5)–(7), with 30 variables bounded in the range [0;1].

$$f_1(\vec{x}) = x_1 \quad (5)$$

$$f_2(\vec{x}) = g(\vec{x}) \times \left[1 - \sqrt{\frac{x_1}{g(\vec{x})}} - \frac{x_1}{g(\vec{x})} \times \sin(10 \times \pi \times x_1)\right] \quad (6)$$

$$g(\vec{x}) = 1 + \frac{9}{n-1} \times \sum_{i=2}^{n} x_i \quad (7)$$

Fig. 2 shows results for the two problems. Convergence is obtained after 20 generations of populations of 30 individuals and after 300 generations with a population size of 40 individuals, for the first and second problem, respectively. In both cases, optimal individuals are well distributed all along a Pareto front. The algorithm is therefore capable to converge while offering diversity among the non-dominated solutions.