BIOINSPIRED FLOW OPTIMIZATION

Petros Koumoutsakos, Stefan Kern and Nikolaus Hansen
Computational Science and Engineering Laboratory,
ETH Zürich
Universitätsrasse 6, Zürich, CH-8092, Switzerland
petros@ethz.ch

ABSTRACT

Bioinpired optimization is concerned with the development and implementation of algorithms and devices based on our experience and understanding of nature. Fluid mechanics are one of the most prominent paradigms of this type of optimization as humanity has been always fascinated from the ways the majestic eagle exploits the wind and the wiggling sperm navigates in the seminal fluid. This fascination has led to mimesis and the development of engineering designs that immitate natural forms and functions. Besides mimicking the final design it is also possible to mimic the processes by which this is achieved leading to genetic algorithms and evolution strategies that can be cast into optimization problems. In this article we discuss bioinspired algorithms for flow optimization by describing some of the fundamental concepts of these techniques and by illustrating their advantages and drawbacks in selected case studies from our research. We discuss single and multi-objective optimization in noisy environments as they pertain to combustion in experimental test-rigs and turbomnachinery, and the use of evolutionary algorithms and local learning models for the optimization of expensive cost functions as applied to simulations of anguiliform swimmers.

INTRODUCTION

Humanity has always sought inspiration from nature in making its devices. Nature evolves its tools over millions of years and we continue to strive to understand the workings of its creations. At the same time humans continue to make tremendous technological progress, enhancing our capabilities to understand nature and to interact effectively (or at times destructively) with our environment.

Fluid mechanics is a key paradigm of our understanding of nature and a prime example of the common approaches and the differences between nature and human beings in utilizing wind and water. Looking at the flight of the mosquito, nature may be attributed the characteristic of a "tinkerer" when looking at all its long legs and fragile wings, whereas engineers continue to develop sophisticated machines such as fighter jets and microfluidic devices. Natural designs, like the one of mosquitos, have survived their environment over hundreds of millions of years while our flight experiments are barely a hundred year old. Yet these six orders of magnitude are not daunting. We may argue that we have technology at our disposal to achieve specific goals that for natural creations may be contradictory to their main dictum of survival. The argument of this author is that we have technology available to "compete" with the magnificent workings of natural creations but that we need to optimize its utilisation. In turn we may seek natural inspiration not only from mimesis, that is making the airplane wings resemble those of a bird, but by also examining the processes by which nature develops its devices.

Bioinspired algorithms for optimization owe their inception to the work of Ingo Rechenberg at TU Berlin in the early 60's. His work as well as other prominent German scientists (such as the architect Otto Frei), [1] of that generation, were influenced by the biology teachings of Johann-Gerhard Helmcke at the Max Planck Institute. The works of Rechenberg and Frei though focusing on seemingly unrelated disciplines (fluid mechanics and architecture) share the same roots of reconsidering the relation between engineering thought and natural processes. In 1964 Rechenberg developed Evolution Strategies (Rechenberg, 1994) by working in the wind tunnel to recover the possibly optimal form of a flat plate bent up in a zig-zag form ("Zickzack-Platte"). The five angles of the corrugated plate were the parameters of the optimization with 0 being the a-priori known answer. Deterministic methods led to local optima while finally the idea of using the "Galtonbrett" to generate random numbers for the plate angles and the use of a systematic evaluation of the better parameters led to a flat plate as the device with the minimum angle.

In the last 40 years bioinspired optimization has received attention by many disciplines from architecture to aerodynamics leading to the immitaion of natural forms and processes. In fluid mechanics bioinpired algoritrhms remain secondary to gradient based techniques although they continue to receive increased attention. Common arguments against the so-called Evolutionary Algorithms are their slow convergence and large numbers of ad-hoc parameters while one of the key arguments of their supporters is that they "work" and the "work well for real problems". In this paper we wish to highlight the advantages and drawbacks of these techniques as they pertain to flow optimization.

Flow optimization involves challenges such as noise, multiple optima, discontinuities, nondifferentiability and high computational or experimental costs of a single objective function evaluation. The cost of optimizing expensive problems is dominated by the number of fitness function evaluations required to reach an acceptable solution.

In recent years algorithmic developments of bioinpired algorithms have provided a mathematical and computational framework for bioinspired optimization algorithms by linking them with machine learning algorithms. Algorithms such as Evolution Strategies and Genetic Algorithms and their numerous variants can be envisioned as randomised algorithms where the optimum parameters are sought by sampling a suitable probability distribution (Kern et al, 2004). The information obtained during the optimization can be exploited in order to enhance the identification of the distribution underlying the optimization problem at hand and increase the performance of the optimization algorithm. Evolutionary algorithms differ in the type and amount of past information they utilize. One can distinguish between

[1] See http://de.wikipedia.org

algorithms that solely use parameter values and rank information of past candidate solutions, and algorithms that additionally use the computed values of the cost function. The algorithms of the first group usually learn probability distributions which are used to sample new candidate solutions. A number of different approaches to the learning of probability distributions evolved from Evolution Strategies (Beyer and Schwefel 2002) and Genetic Algorithms (Mühlenbein and Paass 1996; Pelikan et al. 1999). In a recent study (Kern et al., 2004) we identify important characteristics of different approaches in continuous domain. Explicit model building, regularization, and incorporation of a population independent memory are a few of them. The empirical results indicate that unimodal and non-separable functions are particularly well covered by the CMA-ES approach (Hansen and Ostermeier 2001; Hansen et al. 2003) that uses a memory. The results for multimodal functions are not so clear-cut and reveal a strong influence of the population size on the performance. The CMA-ES has been used in numerous applications ranging from optimization of simulated micorfluidic channels (Müller et al., 2004) to the optimal control of combustion instabilities (Hansen et al., 2007)

The algorithms of the second group build an empirical model (response surface, surrogate models) that approximates the fitness function which then is used in the optimization procedure using different strategies. We distinguish between Evolution Control and the Surrogate Approach. In Evolution Control, a controlled fraction of individuals are evaluated on the expensive fitness function, the remainder only on the model. In the surrogate approach, a fitness function model is constructed for an initial training set of evaluated points. An optimization algorithm then searches for the optimum of the models fitness prediction. The predicted optimum is evaluated on the fitness function and the result of the evaluation is added to the models training data. The procedure then iterates by searching for the optimum on the improved model. In the literature a wide variety of models are used as fitness function models. One particularly promising approach is the use of Gaussian process regression as it provides the key advantage of predicting an uncertainty measure in the form of a standard deviation for the predicted function value. Gaussian Processes have been used for the optimization of compressor blades (Büche et al., 2005) while the use of surrogate models has been introduced for optimization of trailing edge for noise reduction (Marsden et. al 2002). Surrogate modeling for control of trailing aircraft wakes has been reported in (Cottet et al., 2000). For low dimensions the surrogate method clearly outperforms other approaches. For higher dimensions building a reliable surrogate becomes is faced with the "curse of dimensionality" and efficient distribution estimation algorithms are superior in identifying optimal parameters.

We note here that bioinspired techniques should not be considered as competitors of deterministic techniques such as optimal control. Gradient based techniques are well suited to applications where the knowledge of the problem allows a precise mathematical description and they constitute the method of choice in numerous engineering applications. In fluid mechanics however this mathematical description may not be always available, in particular for complex configurations and for processes based on experimental and empirical settings.

The large numbers of iteration often involved in EA's has largely limited their applicability to problems solved through numerical simulations. Recent advances in hardware and the automation of experimental and industrial setups has increased the suitability of these methods to these problems. Experimental setups present a number of challenges to any optimization technique including: availability only of pointwise information, experimental noise in the objective function, uncontrolled changing of environmental conditions and measurement failure.

In this article we present case studies from the development of single and multi-objective optimization algorithms for fluid mechanics problems. Applications such as the optimization of combustion processes in an experimental test rig have guided the development of single and multi-objective evolutionary algorithms capable of handling noise and applications such as the simulation of anguiliform swimmers have guided the development of optimization techniques capable of handling expensive function evaluations.

ALGORITHMS FOR EVOLUTIONARY OPTIMIZATION

We consider a black-box optimization scenario where we want to minimize a single objective function $f : \mathcal{R}^n \rightarrow \mathcal{R}, \boldsymbol{x} \rightarrow f(\boldsymbol{x})$. In this problem the information about the problem is acquired only through the evaluation of the cost function for different parameter settings. Bioinspired algorithms can be cast in the framework of randomised search algorithms by considering the optimization problem as the stochastic search for a parameter vector $\boldsymbol{x}$ that will minimize $f(\boldsymbol{x})$. The search points are then sampled iteratively (over several "generations") from a distribution $P(\boldsymbol{x}|\theta^g)$ with parameters $\theta = \theta^g$ that are being adaptively adjusted using an algorithm $F_\theta(\theta^g, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_\lambda, f_1, \cdots, f_\lambda)$ in each generation based on the information received by the objective function.

1. Initialise parameters: θ^0 and distribution $P(\boldsymbol{x}|\theta^g)$
2. For generation $g = 0, 1, 2, \cdots$
   (a) Sample distribution $P(\boldsymbol{x}|\theta^g) \rightarrow \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_\lambda$
   (b) Evaluate $f(\boldsymbol{x}_1) = f_1, \cdots, f(\boldsymbol{x}_\lambda) = f_\lambda$
   (c) Update $\theta^{g+1} = F_\theta(\theta^g, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_\lambda, f_1, \cdots, f_\lambda)$
   (d) Break, if termination criterion is reached

Randomized algorithms are considered to be robust in rugged objective function landscapes and may handle effectively discontinuities, ridges and local minima. In the following we describe the Covariance MAtrix Adaptation technique and its extension for noisy problems.The CMA-ES has ben shown to perform well in a large number of real-world problems.[2] and it exemplifies several of the aspects of bioinspired optimization algorithms.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) was developed by (Hansen and Ostermeier, 1996; Hansen and Ostermeier, 2001; Hansen et al., 2003; Hansen and Kern 2004) and is the method of choice for several of our flow optimization applications. This choice is dictated by several reasons: CMA-ES is a *non-elitist continuous domain* evolutionary algorithm. Non-elitism avoids systematic fitness overvaluation on noisy objective functions as even solutions with exceptional fitness values survive only one generation. In addition, the selection in CMA-ES is solely based on the ranking of solutions providing additional robustness

[2] See http://www.inf.ethz.ch/personal/hansenn/cec2005.html and http://www.inf.ethz.ch/personal/hansenn/cmaapplications.pdf

in a noisy environment. Finally the CMA-ES provides an effective adaptation of the search distribution to the landscape of the objective function and can be reliably used with small population sizes allowing for a fast adaptation in an online application. The CMA-ES adapts the covariance matrix of a normal search distribution to the given objective function topography achieving nearly optimal covariance matrices for convex-quadratic objective functions. The CMA-ES incorporates **invariance** to a full rank linear transformation of the solution vector. The covariance matrix in the CMA-ES is unbiased, as under random selection the updated covariance matrix equals the previous covariance matrix in expectation.

Following the algorithm in (Hansen et al., 2003) CMA-ES can be formulated as follows: Given an initial mean value $\boldsymbol{m} \in \mathbb{R}^n$, the initial covariance matrix $\boldsymbol{C} = \mathbf{I}$ and the initial step-size $\sigma \in \mathbb{R}_+$, the λ candidate solutions $\boldsymbol{x}_k$ of one generation step obey

$$\boldsymbol{x}_k = \boldsymbol{m} + \sigma \boldsymbol{y}_k, \quad k = 1, \ldots, \lambda, \tag{1}$$

where $\boldsymbol{y}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C})$ denotes a realization of a normally distributed random vector with zero mean and covariance matrix $\boldsymbol{C}$. Equation (1) implements mutation in the EA by adding a random vector. The solutions $\boldsymbol{x}_k$ are evaluated on L and ranked such that $\boldsymbol{x}_{i:\lambda}$ becomes the i-th best solution vector and $\boldsymbol{y}_{i:\lambda}$ the corresponding random vector realization.

The algorithm requires updates of $\boldsymbol{m}$, σ, and $\boldsymbol{C}$ for the next generation step. For $\mu < \lambda$ let

$$\langle \boldsymbol{y} \rangle = \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda}, \quad w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1 \tag{2}$$

be the weighted mean of the μ best ranked $\boldsymbol{y}_k$ vectors. The recombination weights sum to one. The so-called *variance effective selection mass* $\mu_{\text{eff}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \geq 1$ will be used in the following. The mean of the new distribution becomes

$$\boldsymbol{m} \leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{y} \rangle = \sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda}. \tag{3}$$

Equation (3) determines the center of the next population. The equation implements selection by using $\mu < \lambda$. Using *different* recombination weights must also be interpreted as selection mechanism. The equation implements recombination by taking a (weighted) mean of parental solutions.

CMA-ES relies on step-size control the "conjugate" evolution path $\boldsymbol{p}_\sigma \in \mathbb{R}^n$. The evolution path cumulates an exponentially fading pathway of the population mean in the generation sequence. Assuming that the optimal step-size leads to conjugate steps, the length of the conjugate evolution path can be used as adaptation criterion for σ. Initialized with $\boldsymbol{p}_\sigma = \mathbf{0}$ the update of $\boldsymbol{p}_\sigma$ (so-called cumulation) and σ reads :

$$\begin{aligned} \boldsymbol{p}_\sigma \quad &\leftarrow \quad (1 - c_\sigma)\, \boldsymbol{p}_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}\, \boldsymbol{C}^{-\frac{1}{2}} \langle \boldsymbol{y} \rangle && (4) \\ \sigma \quad &\leftarrow \quad \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\boldsymbol{p}_\sigma\|}{\widehat{\chi}_n} - 1 \right) \right) && (5) \end{aligned}$$

where $1/c_\sigma > 1$ determines the backward time horizon of the evolution path $\boldsymbol{p}_\sigma$, damping $d_\sigma \approx 1$ controls the change magnitude of σ, and $\widehat{\chi}_n$ is the expected length of a random variable distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The evolution path is appropriately normalized. We have $\boldsymbol{C}^{-\frac{1}{2}} \overset{\text{def}}{=} \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{B}^{\mathrm{T}}$, where $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{D}^2\boldsymbol{B}^{\mathrm{T}}$ is an eigendecomposition of the symmetric, positive definite covariance matrix $\boldsymbol{C}$.[3] The transformation $\boldsymbol{C}^{-\frac{1}{2}}$ rescales $\langle \boldsymbol{y} \rangle$ into an isotropic reference system. Given $\boldsymbol{y}_{i:\lambda}$ distributed according to $\mathcal{N}(\mathbf{0}, \boldsymbol{C})$, as under random selection, we can derive that $\sqrt{\mu_{\text{eff}}}\boldsymbol{C}^{-\frac{1}{2}} \langle \boldsymbol{y} \rangle$ is distributed according to $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The transformations make the expected length of $\boldsymbol{p}_\sigma$ independent of its orientation and allow the comparison of the length of $\boldsymbol{p}_\sigma$ with its expected length $\widehat{\chi}_n$ in (5). Step-size σ is increased if and only if $\|\boldsymbol{p}_\sigma\| > \widehat{\chi}_n$, and decreased if and only if $\|\boldsymbol{p}_\sigma\| < \widehat{\chi}_n$. In practice we use the approximation $\widehat{\chi}_n = \sqrt{2}\,\Gamma(\frac{n+1}{2})/\Gamma(\frac{n}{2}) \approx \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$.

Similar to (4) an evolution path $\boldsymbol{p_c}$ is constructed to update the covariance matrix. The covariance matrix admits a rank-one and a rank-μ update.

$$\begin{aligned} \boldsymbol{p_c} \quad &\leftarrow \quad (1 - c_{\boldsymbol{c}})\, \boldsymbol{p_c} + h_\sigma \sqrt{c_{\boldsymbol{c}}(2 - c_{\boldsymbol{c}})\mu_{\text{eff}}}\, \langle \boldsymbol{y} \rangle && (6) \\ \boldsymbol{C} \quad &\leftarrow \quad (1 - c_{\text{cov}})\boldsymbol{C} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \underbrace{\boldsymbol{p_c}\boldsymbol{p_c}^{\mathrm{T}}}_{\text{rank-one update}} \\ &\qquad + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \underbrace{\sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda} \boldsymbol{y}_{i:\lambda}^{\mathrm{T}}}_{\text{rank-}\mu\text{ update}} && (7) \end{aligned}$$

where $c_{\text{cov}} \leq 1$ is a learning rate, $\mu_{\text{cov}} \geq 1$ determines the portion between rank-one and rank-μ updates, and $h_\sigma = 0$ if $\|\boldsymbol{p}_\sigma\| > \left(1.5 + \frac{1}{n-0.5}\right) \widehat{\chi}_n \sqrt{1 - (1 - c_\sigma)^{2(g+1)}}$, and 1 otherwise, where g is the generation counter. Consequently, the update of $\boldsymbol{p_c}$ is stalled whenever $\boldsymbol{p}_\sigma$ is considerably longer than expected. This mechanism is decisive after a change in the environment which demands a significant increase of the step-size whereas fast changes of the distribution shape are postponed until after the step-size is increased to a reasonable value. A detailed discussion of the strategy parameters can be found in (Hansen and Ostermeier, 2001; Hansen et al., 2003; Hansen and Kern, 2004).

Noise-Handling CMA-ES

In order to handle noise high noise levels in the fitness function we develop a Noise-Handling CMA-ES (NH-CMA-ES). The algorithm is based on a $(\mu/\mu, \lambda)$ CMA-ES with the default parameters from (Hansen and Kern, 2004). The noise handling preserves all invariance properties of the CMA-ES, but biases the population variance when too large a noise level is detected.

The noise measurement is based on measured *rank changes induced by reevaluations* of solutions. The algorithm outputs a noise measurement value s and reads

1. Set $L_i^{\text{new}} = L_i^{\text{old}} = L(\boldsymbol{x}_i)$, for $i = 1, \ldots, \lambda$, and let $\mathcal{L} = \{L_k^{\text{old}}, L_k^{\text{new}} | k = 1, \ldots, \lambda\}$, where λ is the number of offspring in the CMA-ES.

2. Compute λ_{reev}, the number of solutions to be reevaluated; $\lambda_{\text{reev}} = f_{\text{pr}}\,(r_\lambda \times \lambda)$ where the function $f_{\text{pr}} : \mathbb{R} \to \mathbb{Z},\ x \mapsto \begin{cases} \lfloor x \rfloor + 1 & \text{with probability } x - \lfloor x \rfloor \\ \lfloor x \rfloor & \text{otherwise} \end{cases}$. If $r_\lambda \times \lambda < 1$ and $\lambda_{\text{reev}} = 0$ for more than $2/(r_\lambda \times \lambda)$

[3] Columns of $\boldsymbol{B}$ are an orthonormal basis of eigenvectors, $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}} = \mathbf{I}$. Diagonal elements of the diagonal matrix $\boldsymbol{D}$ are square roots of the corresponding positive eigenvalues. The matrix $\boldsymbol{D}$ can be inverted by inverting its diagonal elements. From these definitions it follows that $\boldsymbol{y}_k \sim \sigma\boldsymbol{B}\boldsymbol{D}\mathcal{N}(\mathbf{0}, \mathbf{I})$ which allows the generation of the random vector realizations on the computer.

generations, set $\lambda_{\text{reev}} = 1$ to avoid too long sequences without reevaluation.

3. Reevaluate solutions. For each solution $i = 1, \ldots, \lambda_{\text{reev}}$ (assuming the solutions of the population are i.i.d., we can choose *the first* λ_{reev} solutions for reevaluation)

   (a) Apply a small perturbation: $\boldsymbol{x}_i^{\text{new}} = \texttt{mutate}(\boldsymbol{x}_i, \varepsilon)$ where $\boldsymbol{x}_i^{\text{new}} \neq \boldsymbol{x}_i \iff \varepsilon \neq 0$. For the CMA-ES we might apply $\texttt{mutate}(\boldsymbol{x}_i, \varepsilon) = \boldsymbol{x}_i + \varepsilon\sigma\mathcal{N}(\mathbf{0}, \boldsymbol{C})$, where $\mathcal{N}(.)$ denotes a multi-variate normal distribution and σ and $\boldsymbol{C}$ are the step-size and the covariance matrix from the CMA-ES.

   (b) Reevaluate the solution: $L_i^{\text{new}} = L(\boldsymbol{x}_i^{\text{new}})$

4. Compute the rank change Δ_i. For each chosen solution $i = 1, \ldots, \lambda_{\text{reev}}$ the rank change value, $\Delta_i \in \{0, 1, \ldots, 2\lambda - 2\}$, counts the number of values from the set $\mathcal{L} \setminus \{L_i^{\text{old}}, L_i^{\text{new}}\}$ that lie between L_i^{old} and L_i^{new}.

5. Compute the noise measurement, s. Therefore the rank change value, Δ_i, is compared with a limit $\Delta_\theta^{\text{lim}}$. The limit is based on the distribution of the rank changes on a random function L and the parameter θ (see text).

6. Re-rank the solutions according to their rank sum, *i.e.* $\texttt{rank}(L_i^{\text{old}}) + \texttt{rank}(L_i^{\text{new}})$. Ties are resolved first using the absolute rank change $|\Delta_i|$, where the mean $\Delta_i = \frac{1}{\lambda_{\text{reev}}} \sum_{j=1}^{\lambda_{\text{reev}}} |\Delta_j|$ is used for solutions $i > \lambda_{\text{reev}}$ not being reevaluated, and second using the (mean) function value.

The parameters are set to $r_\lambda = \max(0.1, \frac{2}{\lambda})$, $\varepsilon = 10^{-7}$, and $\theta = 0.2$.

Two noise treatments are used in NH-CMA-ES. First, increase of the evaluation (measuring) time (or reevaluations of the fitness function). Second, increase of the population variance (step-size σ), that can have three beneficial effects. (a) the signal-to-noise ratio is likely to improve, because the population becomes more diverse; (b) the population escapes search-space regions with too low a signal-to-noise ratio, because in these regions the movement of the population is amplified; and (c) premature convergence is prevented. The noise treatment algorithm applied after each generation step uses noise measurement s, and affects step-size σ and evaluation time t_{eval}.

```
$\bar{s} \leftarrow (1 - c_s)\,\bar{s} + c_s s$
if $\bar{s} > 0$        % apply noise treatment
    if $t_{\text{eval}} = t_{\max}$
        $\sigma \leftarrow \alpha_\sigma \sigma$
    $t_{\text{eval}} \leftarrow \min(\alpha_t t_{\text{eval}}, t_{\max})$
else if $\bar{s} < 0$   % decrease evaluation time
    $t_{\text{eval}} \leftarrow \max(t_{\text{eval}}/\alpha_t, t_{\min})$
```

Typical parameter settings can be found in (Hansen et al., 2007)

Local meta model CMA-ES

Local meta-models can be used to enhance the efficiency of CMA-ES in the optimization of computationally expensive problems, resulting in the *local meta-model CMA-ES* (lmm-CMA) (Kern et al., 2006). Locally weighted regression (Atkeson et al., 1997) is used to fit past evaluations of the fitness function stored in a database only in a region around the location of the new parameter vector $\boldsymbol{x}$ to be evaluated. The local models are built consecutively as queries need to be answered and therefore are intrinsically designed for growing training data sets as they occur in the course of an optimization.

For every offspring to be predicted an individual model is built. Given a set of points $(\boldsymbol{x}_j, y_j), j = 1, \ldots, m$, the training criterion C is minimized w.r.t. the parameters $\boldsymbol{\beta}$ of the local mode $\hat{f}$ at query point $\boldsymbol{q}$ and can be written as

$$C(\boldsymbol{q}) = \sum_{j=1}^{m} \left[(\hat{f}(\boldsymbol{x}_j, \boldsymbol{\beta}) - y_j)^2 K\left(\frac{d(\boldsymbol{x}_j, \boldsymbol{q})}{h}\right) \right], \qquad (8)$$

where $K(.)$ is the kernel weighting function, $d(\boldsymbol{x}_j, \boldsymbol{q})$ the distance between data point $\boldsymbol{x}_j$ and $\boldsymbol{q}$, and h is the (local) bandwidth (cf. Kern et al., 2006). Investigations with models $\hat{f}$ of different complexity revealed that local quadratic meta models are preferable. Lower order models were not capable of enhancing the performance of the lmm-CMA. For the calculation of $d(\boldsymbol{x}_j, \boldsymbol{q})$ we propose to utilize the metric of the search distribution of the EA. Evolution strategies as the CMA-ES adapt a multivariate Gaussian mutation distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ to the (local) topography of the function, and the covariance matrix $\boldsymbol{C}$ naturally defines a metric that can be exploited in the calculation of d as fully weighted Euclidean distance

$$d(\boldsymbol{x}_j, \boldsymbol{q}) = \sqrt{(\boldsymbol{x}_j - \boldsymbol{q})^T \boldsymbol{C}^{-1} (\boldsymbol{x}_j - \boldsymbol{q})}. \qquad (9)$$

Because the density of the data points collected in the course of an optimization run changes considerably, an adaptive choice of the bandwidth h is essential. We use a *nearest neighbor bandwidth selection*, where h is set to the distance of the kth nearest neighbor data point to $\boldsymbol{q}$ and thus the volume increases and decreases in size according to the density of nearby data. In this way changes in scale of the distance function d are canceled by the choice of h, giving a scale invariant distribution of the weights to the data. k is set to twice the number of free parameters of the local model being $k = n(n + 3) + 2$ for the local quadratic models.

In lmm-CMA the local meta-models are utilized in the framework of Evolution Control. The fraction of candidate solutions $\boldsymbol{x}$ predicted using the meta-model is dependent on the quality of the model. An elegant way to control model quality without knowing the correct ranking of the complete population is the *approximate ranking procedure* (Runarsson, 2004): In every generation, the offspring are successively evaluated and added to the training set of the fitness function model until the (deterministic) model based selection of the parents remains unchanged in two consecutive iteration cycles. This results in an adaptive control mechanism determining the number of evaluated individuals in every generation. The approximate ranking procedure for a $(\mu/\mu, \lambda)$-lmm-CMA reads

1. *approximate*: build $\hat{f}(\boldsymbol{x}_k), k = 1, \ldots, \lambda$ based on evaluations in training set $\mathcal{S}$

2. *rank*: based on $\hat{f}$ generate ranking$_0^\mu$ of the μ best individuals

3. *evaluate*: n_{init} best individuals based on $\hat{f}$, add to $\mathcal{S}$

4. **for** $i := 1$ **to** $(\lambda - n_{\text{init}})/n_b$ **do**

   (a) *approximate*: build $\hat{f}(\boldsymbol{x}_k), k = 1, \ldots, \lambda$ based on $\mathcal{S}$

   (b) *rank*: based on $\hat{f}$ generate ranking$_i^\mu$ of the μ best individuals

(c) **if** ($\text{ranking}^{\mu}_{i-1} == \text{ranking}^{\mu}_{i}$): **goto** 5
else: *evaluate:* n_b next best unevaluated points based on $\hat{f}$, add to $\mathcal{S}$

5. **if** ($i > 2$): $n_{\text{init}} = \min(n_{\text{init}} + n_b, \lambda - n_b)$

6. **if** ($i < 2$): $n_{\text{init}} = \max(n_b, n_{\text{init}} - n_b)$

n_{init} is the number of initial evaluations performed before the model iteration loop is entered.

Multi-Objective Evolutionary Algorithms (MOEA)

A multi-objective optimization problem can be described by an objective vector f and a corresponding set of design variables x. Without loss of generality we can consider the minimization of f. Formally:

$$\begin{aligned} \min f(x) &= (f_1(x), f_2(x), \ldots, f_m(x)) \in F \\ \text{where } x &= (x_1, x_2, \ldots, x_n) \in X, \end{aligned} \quad (10)$$

where $X \in \mathcal{R}^n$ is the n-dimensional design space, $F \in \mathcal{R}^m$ is the m-dimensional objective space. Here both the design and objective space are real spaces, as they correspond to continuous variables and measured objectives for the proposed application. A partial ordering can be applied to solutions in the objective space F by the dominance criterion. A solution a in X is said to dominate a solution b in X ($a \succ b$), if it is superior or equal in all objectives and at least superior in one objective. This is expressed as:

$$\begin{aligned} a \succ b, \text{ if} \qquad & \forall\, i \in \{1, 2, \ldots, m\}: \quad f_i(a) \le f_i(b) \;\wedge \\ & \exists\, j \in \{1, 2, \ldots, m\}: \quad f_j(a) < f_j(b) \end{aligned} \quad (11)$$

The solution a is said to be indifferent to a solution c, if neither solution is dominating the other one. When no a priori preference is defined among the objectives, dominance is the only way to determine, if one solution performs better than the other (Fonseca and Fleming, 1995). The complete set of Pareto ideal solutions represents the best solutions to a problem. Starting from a Pareto solution, one objective can only be improved at the expense of at least one other objective. From the Pareto definition, two issues have to be considered by the formulation of an evolutionary optimization algorithm for Pareto optimization. On one hand, the algorithm must be able to converge sufficiently fast towards the Pareto front, while on the other, it must preserve diversity among its population in order to be able to spread over the whole Pareto front. A common difficulty is the focusing of the population on a certain part of the Pareto front, which is known as genetic drift. In single objective optimization this is unimportant, since convergence to a single (global) optimum is desired.

For a multi-objective problem, a *selection operator* selects in average the less dominated solutions from P and places them in a parent population P_p of size μ. The *recombination operator* chooses randomly individuals from the parent population P_p and recombines them into a child. For the *mutation operator*, the variables of a child are mutated by adding normally distributed random numbers. A termination criterion for the evolution may be the maximal allowed number of generations.

The Strength Pareto Evolutionary Algorithm (SPEA) of Zitzler and Thiele (1999) is a well-established Pareto-optimization algorithm. The algorithm entails a fitness assignment and selection mechanism based on the concept of elitism. SPEA uses the nondominated solutions for the fitness assignment. First, the fitness of each nondominated solution is computed as the fraction of the population, which it dominates. The fitness of a dominated individual is equal to one plus the fitness of each nondominated solution by which it is dominated. This fitness assignment guarantees that the fitness of nondominated solutions is always lower than the fitness of the dominated.

Elitism is a technique of preserving always the best solutions obtained so far. In multi-objective optimization, elitism is performed by storing the nondominated solutions in an archive.In the selection process individuals of the current population and of the archive are competing in a binary tournament where contrary to the standard tournament selection the solution with the lower fitness wins.

In order to preserve diversity in the archive and to keep its size limited, a clustering algorithm is used. Clustering removes solutions in areas of high density as measured in the objective space.

The studies of Zitzler and Thiele (1999) have illustrated that elitism improves the performance of multi-objective evolutionary algorithms on noise-free test problems. Elitism is inserting nondominated solutions in the selection process, and thus increasing the selection pressure. Some researchers state elitism as a necessity for multi-objective optimization (Van Veldhuizen and Lamont, 2000) since information may be lost by the stochastic selection operator. However, this advantage is debatable for noisy objective functions.

MOEA for Noisy Applications

For optimization noisy applications like real-world problems and experimental setups, modifications are needed to the standard multi-objective evolutionary algorithms in order to increase their robustness. We propose three modifications for an extended multi-objective algorithm for noisy environments:

1. *Domination dependent lifetime:* In contrast to elitism, which may preserve nondominated solutions for an infinite time, a lifetime κ is assigned to each individual. The lifetime is shortened, if the solution dominates a major part of the archive. This limits the impact of a solution and safeguards against outliers.

2. *Re-evaluation of solutions:* It is common to delete solutions with expired lifetime. We re-evaluate archive solutions with expired lifetime and add them to the population. This allows good solutions to stay in the evolutionary process, but their objective values will change due to the noise in the re-evaluation.

3. *Extended update of the archive:* The SPEA algorithm updates the archive always by adding the current population to the archive and removing the dominated solutions. We extend the update to *all* solutions with non-expired lifetime. This hinders loss of information, since solutions which were removed by clustering or domination may reenter the archive.

With these features NT-SPEA uses the advantage of an archive as convergence accelerator, but it reduces the risk induced by outliers.

The dominance-dependent lifetime reduces the impact of a solution. An individual that dominates a large fraction of the archive has a high chance of being selected in the selection process, but is assigned the shortest lifetime.

While the principle of limited lifetime is a key element to remove outliers, the re-evaluation allows good solutions to stay in the selection process by re-entering the archive. In the case of an outlier, it is not likely, that the re-evaluated copy is again an outlier with good objective values and hence it would not re-enter the archive. On the other hand, solutions with good design variable settings are likely be nondominated again, if the effect of noise is limited.

The extended update considers the nondominated among all solutions with non-expired lifetime for the update of the archive. Since the assigned lifetime differs between the solutions, the set of nondominated solutions changes. Dominated become nondominated, if the lifetime of their dominator expires. This is especially important if a noisy solution or an outlier dominates a large fraction of the archive. The dominated are then removed from the archive. The noisy solution or outlier is assigned a short lifetime. After the lifetime expires the removed nondominated may be re-selected to the archive. With the original update of SPEA, their information is lost. The Noise-Handling SPEA is given by:

```
1. Generate an initial population $P$ and an empty archive $A$.
2. Define a maximal lifetime $\kappa_{max}$ for individuals (in generations).
3. Evaluate the objectives of the individuals in $P$.
4. **while** termination criterion is not fulfilled **do**
   (a) Assign lifetime: Compute for each individual in $P$ the fraction of the archive that it dominates. The lifetime $\kappa$ of the individual is inverse proportional to the fraction
   (b) Update $A$: Remove all solutions from $A$ and refill it with all solutions, whose lifetime is not expired.
   (c) Then remove all dominated solutions.
   (d) Limit the size of $A$ by clustering.
   (e) Fitness assignment: Assign fitness to the individuals in $P$ and $A$.
   (f) Selection: Use tournament selection for selecting the parent population $P_p$ from $P \cup A$.
   (g) Recombination: Generate a new population $P$ by recombination of the individuals in $P_p$.
   (h) Mutation: Mutate the individuals in $P$.
   (i) Re-evaluation: Select the solutions from $A$ with expiring lifetime and add a copy for re-evaluation to the population $P$
   (j) Evaluate the objectives of the individuals in $P$.
   (k) **end while**
```

A convergence comparison for various implementations of SPEA has been performed on noisy and noise-free test functions. In general, a decrease in convergence is observed when noise is introduced. The concept of elitism is analyzed in the presence of noise. In the absence of noise, elitism can be used as a convergence accelerator. However, for different types of noise, elitism can imply a significant disadvantage, since the optimization can get misled by outliers.

The NT-SPEA overcomes the problem by introducing dominance-dependent lifetime and accelerates the convergence by using an archive. The archive is modified by the re-evaluation of nondominated solutions and an extended update. For the noise-free test problems, NT-SPEA shows similar convergence to the original SPEA, which converges best. This is a major advantage compared to a non-elitistic and a statistical implementation of SPEA and the ESPEA of Teich.

While NT-SPEA performs equal or superior to the best of the other implementations for problems with normally distributed noise, it clearly outperforms all algorithms for problems with outliers. A further advantage is that NT-SPEA can handle moving optima over time or changing environmental conditions. The algorithm re-evaluates solutions after a limited lifetime, therefore adapts the objective values according to the changing values.

The algorithm is successfully applied to an automated optimization of gas turbine burners. The process produces in an automated fashion an experimental nondominated front for minimizing pulsation and emissions of an industrial burner. Automated optimization can be considered a supporting tool in the design process, complementing physical understanding as well as trial-and-error design.

OPTIMIZATION UNDER UNCERTAINTY

In experiments and industrial configurations we often detect different results for repeated measurements of the same operating point. The differences are attributed to noise and unobserved factors in the setup.

Noise may occurs in various areas in the experiment: The setting of the operating conditions is within a limited precision. In the realization, the operating condition may vary over time and finally measurement errors occur. It is up to the careful setup by the experimenter to keep the noise within a limited range. We define this noise, which is present in all measured experiments, as *experimental noise*. It is often modeled by a normal distribution with defined mean and standard deviation, which define a priori knowledge of the processes involved.

In addition, during an automated optimization cycle, an experimental measurement may fail completely, producing *outliers*, i.e. arbitrary nonphysical results. This occurs very rarely, but may have large impact on the automated process optimization if not recognized by a supervisor or captured by some penalty function. Outliers cannot be described by a statistical model with given mean and deviation, but are best modeled by a probability of occurrence. Noise and outliers influence the multi-objective optimization process by misleading the selection operation. Hence unrealistic inferior solutions may dominate superior ones, thus delaying or completely misleading the convergence to an unrealistic Pareto front.

Evolutionary Optimization of Experiments

Evolution Strategies (ES) were initially developed for experiments four decades ago (see e.g. Box (1957) or Schwefel (1977)). Early studies have shown that ES sample more efficiently the search space, and that is what makes them better suited for experimental purposes than grid search methods (Box and Wilson (1951)). Applying evolution to experiments implies coping with additional effects that are usually not present in simulations. These effects include measurement uncertainties, digital signal quantification and uncontrollable environmental influences.

Measurement errors can have a large impact on selection operators in EAs. When the difference between two fitness values is less than the measurement uncertainty, it is possible that the "better" individual only has a better fitness value due to measurement errors and not due to its parameter set which could actually be worse than others. To avoid this problem, mutation step sizes must always be larger then any uncertainty in order to get changes in the objective function that are larger then the general measurement noise. This poses a lower limit to the convergence of the evolution strategy and an additional constraint to the method of step size adaptation, to ensure that step sizes are always significant. ES using recombination are more robust against distorted selection and should therefore be favored for experimental purposes. Digital hardware is capable of producing or measuring signals only in *discrete steps*. The optimization of acoustically driven blooming jets (Sbalzarini et al. 2001) identified that for certain hardware such as onboard wave form generators, these steps can be as large as 0.5% of full range, depending on the buffer size of the device. For a wave form generator, this could mean, that parameter vectors differing less than one such discrete step will cause the same physical output to be produced. Besides adding another lower limit for the mutation step size of the ES, this quantification also calls for a feedback mechanism for the ES with the values that have actually been produced instead of the parameter vector requested. This requires robustness of the ES against unpredictable changes in the parameter vectors and dictates changes in the CMA-ES (leading to NH-CMA-ES) that keeps track of the evolution path in order to improve convergence.

NH-CMA ES for the optimization of combustion instabilities

The NH-CMA-ES has been used for the system identification and the online optimization of feedback controllers applied to combustion systems (Hansen et al., 2007). The algorithm is applied to gas turbine combustors that are susceptible to thermoacoustic instabilities resulting in imperfect combustion and decreased lifetime. In order to mitigate these pressure oscillations, feedback controllers sense the pressure and command secondary fuel injectors. The controllers are optimized online with the NH-CMA-ES capable of handling noise associated with the uncertainties in the pressure measurements. The present method is independent of the specific noise distribution and prevents premature convergence of the evolution strategy. NH-CMA-ES needs only two additional function evaluations per generation and is therefore particularly suitable for online optimization.

A lab scale test rig is used for the experiments. Preheated air premixed with natural gas flows into a downscaled model for the ALSTOM environmental (EV) swirl burner that stabilizes the flame in recirculation regions near the burner outlet plane. The pressure signal is detected by a water-cooled microphone placed 123 mm downstream of the burner. A MOOG magnetostrictive fuel injector installed close to the flame is used as control actuator. Two controller types are investigated: a simple phase-shift or Gain-Delay controller where gain and delay are optimized by the evolutionary algorithm; and a model-based robust $\mathcal{H}_\infty$ controller where a frequency shift, gain and delay of a previously designed $\mathcal{H}_\infty$ controller (Skogestad and Postlethwaite, 1996; Niederberger et al., 2005) are optimized by the evolutionary algorithm.

The cost function to be minimized is the equivalent continuous level of the sound pressure $L_{eq} = 10\log_{10}(p_s^2)_{av}/p_{\text{ref}}^2$, where $(p_s^2)_{av}$ is the mean squared pressure and $p_{\text{ref}} = 20\mu\text{Pa}$ is the reference pressure. The sound pressure level L_{eq} is acquired from a measurement during t_{eval} seconds with a given controller parameter setting. Spectra achieved with the optimized Gain-Delay and $\mathcal{H}_\infty$ controllers are compared to the uncontrolled plant in Fig 1. They are shown for the plant which has been running for several hours and is thus heated up. For $\lambda = 1.875$ (left) the L_{eq} of the uncontrolled plant is 148.72 dB, the Gain-Delay controller reduces it to 146.67 dB, while the $\mathcal{H}_\infty$ controller reaches 146.16 dB, which is about 15% less. For $\lambda = 2.1$ the values of L_{eq} are 159.87 dB, 147.48 dB and 147.35 dB, respectively. Here the $\mathcal{H}_\infty$ controller performs only slightly better than the Gain-Delay controller, but the control signal contains about 10% less energy.

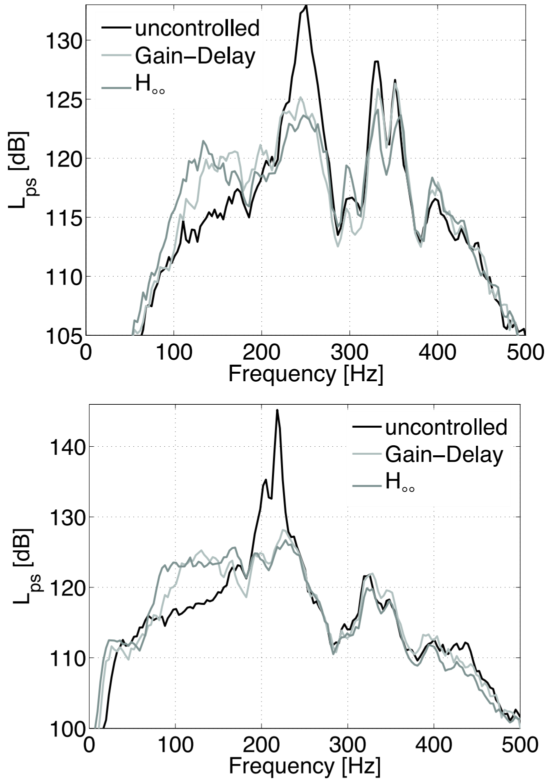

Figure 1: Comparison of the pressure spectra for the uncontrolled, Gain-Delay controlled and $\mathcal{H}_\infty$ controlled plant. Both controllers are NH-CMA-ES optimized. Top: $\lambda = 1.875$, bottom: $\lambda = 2.1$.

The experiments show that the algorithm can optimize different controller types and can cope with changing operating conditions and high levels of noise. Model-based $\mathcal{H}_\infty$ controllers perform best, and can be improved further through the use of the NH-CMA-ES. The optimized solutions deviate remarkably from the originally designed solutions and can make up for uncertainties in the model-building and design process, as well as for time-varying plant characteristics.

NH-SPEA for multiobjective optimization of turbomachinery

The NH-SPEA is applied to the multiobjective optimization of the combustion process of a stationary gas turbine in an industrial setup (Büche et al., 2004). The Pareto front is constructed for the objectives of minimization of NO$_x$ emissions and reduction of the pressure fluctuations (pulsation) of the flame. Both objectives are conflicting affecting the environment and the lifetime of the turbine, respectively. We consider the optimization of a single burner in an atmo-

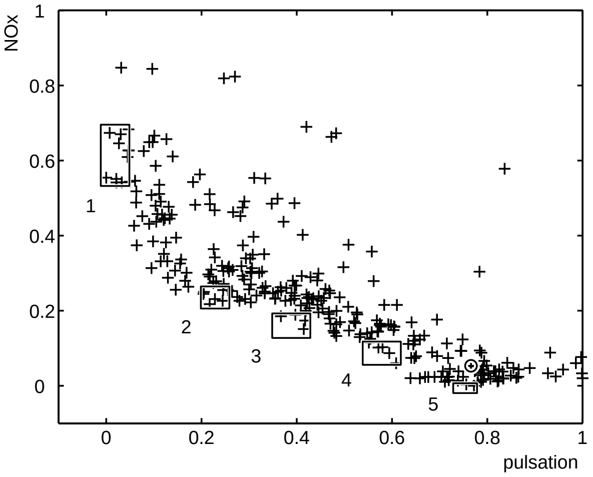

Figure 2: All measured solutions of the burner optimization run [plus symbol] and given standard burner configuration [circular symbol]. 5 boxes mark different areas along the nondominated front.

spheric test-rig. Preheated air enters the test-rig from the plenum chamber and is mixed with fuel in the low-emission burner by swirl. The burner stabilizes the combustion flame in a predefined combustion area by a controlled vortex breakdown. The fuel is natural gas or oil and is injected through injection holes, which are uniformly distributed along the burner. We consider a passive control mechanism, choosing the fuel flow rates through the injection holes of the burner as design variables of the setup, due to the low modification cost for the gas turbine compared to an active control system. 8 continuous valves $V_{i,i=1,\ldots,8}$ are used to control the fuel rates. Each valve V_i controls the mass flow $\dot{m}_i$ through a set of adjacent injection holes along the burner axis.

The optimization is performed using NT-SPEA with a population and archive size of 15 and evaluating a total of 326 different burner settings within one working-day. All solutions are plotted in Fig. 2 in order to show the possible decrease in NO$_x$ emissions and pulsations by the optimization compared to the given standard burner configuration and between the best and worst designs.

The given standard burner configuration is marked in the figure and represents a setting with equal mass flow through all valves. Some solutions found by the optimization process dominate the standard configuration, i.e. are superior in both objectives. Thus the optimization run is successful, delivering improved solutions for both objectives. The occurrence of a wide nondominated front underlines the conflict in minimizing both objectives and just (Pareto) compromise solutions can be found.

EVOLUTIONARY OPTIMIZATION FOR SIMULATIONS

Evolutionary Optimization is well suited for the optimization of flows that can be effectively simulated using Direct Numerical Simulations. Care must be exercised when the simulations involve uncertainties due to parameters such as turbulence models as these uncertainties may interfere with the optimization parameters. In addition as evolutionary algorithms require relatively large numbers of iterations novel optimizations algorithms (such as Local Learning CMA-ES) are required in order to develop effective optimization procedures. An attractive aspect of evolutionary algorithms is that they can be adopted as an optimization wrapper to many flow solvers and empirical calculations. This property in conjunction with their ineherent parallelism provides us with a robust optimization tool.

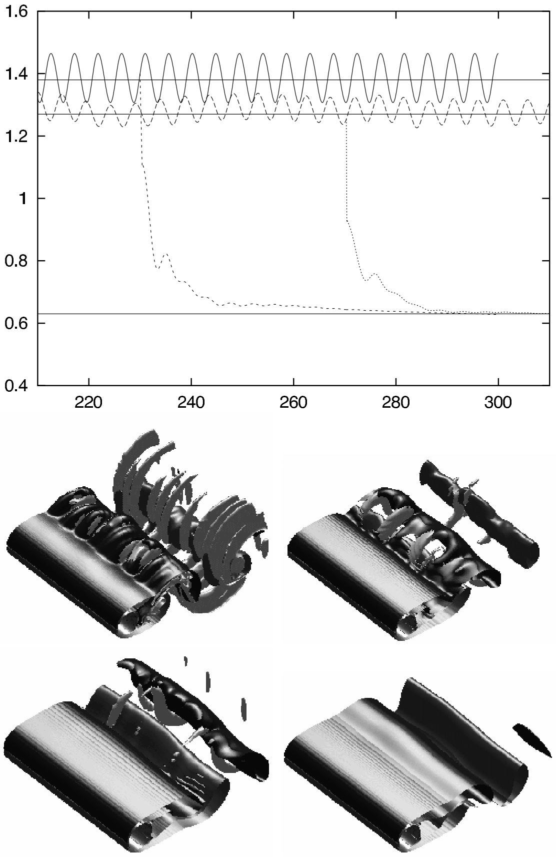

Figure 3: Cylinder drag reduction at Re=300 using Evolution strategies. Top: Drag coefficients for 2D without control (——), 2D with control (- - -), 3D without control (– – –), and3D with control (· · · ·). Bottom: Snapshots of flow past a cylinder illustrating how the three-dimensionality of the flow is killed after control is switched on.

We illustrate some of these issues through two applications: the control of cylinder wakes leading to drastic flow and drag modifications and the optimization of anguilliform swimmers leading to different swimming patters for the same form having different objectives.

Cylinders

A real coded genetic algorithm is implemented for the optimization of actuator parameters for cylinder drag minimization (Milano and Koumoutsakos 2002). We consider two types of idealized actuators that are allowed either to move steadily and tangentially to the cylinder surface (belts), or to steadily blow/suck with a zero net mass constraint. The genetic algorithm we implemented has the property of identifying minima basins, rather than single optimum points. The knowledge of the shape of the minimum basin enables further insights in the system properties and provides a sensitivity analysis in a fully automated way. The drag minimization problem is formulated as an optimal regulation problem. By means of the clustering property of the present genetic algorithm, a set of solutions producing drag reduction of up to 50% is identified. A comparison between the two types of actuators, based on the clustering property of the algorithm indicates that blowing/suction actuation parameters are associated with larger tolerances when compared to optimal parameters for the belt actuators. The possibility to use few strategically placed actuators in order to obtain a significant drag reduction was explored using the clustering diagnostics of this method.

In a related work (Poncet et al. 2005) the optimal belt-actuator parameters obtained by optimizing the two-dimensional case have been employed in three dimensional simulations, by extending the actuators across the span of the cylinder surface. The three dimensional controlled flow exhibits a remarkable two-dimensional character near the cylinder surface, resulting in significant drag reduction.

Optimization of Anguiliform Swimmers

Aquatic animals are smart swimmers achieving remarkable propulsive efficiency. Anguilliform locomotion is a particular case as the animals propel themselves forward by propagating waves of curvature backwards along the body. The lamprey is an animal that employs this primitive (in an evolutionary sense) type of locomotion and has been a popular object to study neural control of locomotion of aquatic animals (Ekeberg 1993; Ijspeert and Kodjabachia 1999). These studies however use strongly simplified models of the fluid dynamics of the fish to learn the motion patterns. In contrast, there exist only few studies computing fully three dimensional viscous flow around swimming creatures, and all of them use apriori defined motion patterns without any control or learning mechanisms (Liu and Kawachi 1999; Zhu et al. 2002; Ramamurti et al. 2002). In our ongoing work we combine control and learning of motion patterns with a fully three dimensional instationary viscous flow simulation of the creature freely moving in the water. The three-dimensional geometry of the anguilliform swimmer is constructed from spatially varying ellipsoid cross sections. The motion pattern of the body is described by a traveling wave of curvature of the mid-line of the body parameterized with 6 parameters.

Two different objectives have been optimized that may correspond to biological functions such as hunting/escaping (for the burst velocity) or migrating (for the efficient swimming). We used the classical CMA-ES and the lmm-CMA for to solve this optimization problem. The wake structures of the present simulations are consistent with several experimental observations. The fast and the efficient swimming mode both shed a double row of vortex rings responsible for the strong lateral jets observed in the wake and shown in Fig. 4. The results provide quantification of the vortex formation and shedding processes and enable the identification of the portions of the body that are responsible for the majority of thrust in anguilliform swimming. In burst swimming the tail is responsible for the majority of the thrust, while in efficient swimming the anterior part of the body also contributes to the thrust.

SUMMARY

Since the doomed flight of Icarus there has been significant progress in the bioinspired optimization of engineering devices utilizing flows for their function. The development of bioinspired algorithms in conjunction with the technological advances of our times provides us with tremendous opportunities for the innovative exploration of flow phenomena. The present article includes a summary of key algorithms and certain applications from our research activities to illustrate that bioinpisred algorithms could complement well expensive flow simulations as well as noisy experiments.

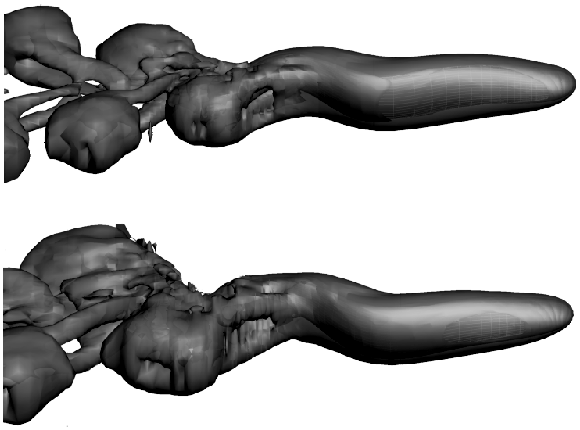

Figure 4: Self propelled anguilliform swimmers optimized for efficient swimming (top) and fast swimming (bottom). Isosurfaces of vorticity magnitude show the characteristic structure of the wake largely consisting of a double row of vortex rings aligned in swimming direction.

Fluid mechanics was responsible for the generation of modern day evolutionary algorithms that today permeate areas from computer science and evolvable hardware to architectural design and micorfluidic chips. We believe that this dynamic interaction of fluids and bioinspired optimization will continue to provide future generations of scientists with a wealth of optimized flows to explore and understand and with improved designs that may not always rely on a complete human understanding or predetermination. Humanity may soon interact with nature beyond the level of mimesis leading to further insight and respect for it's workings.

REFERENCES

Atkeson, C.G., Moore, A.W., and Schaal, S., 1997, "Locally weighted learning", Artificial Intelligence Review 11(1-5), pp. 11–73.

Annaswamy, A. and Ghoniem, A., 2002, "Active control of combustion instabilities: Theory and practice", IEEE Control Systems Magazine 22(6), 37–54.

Beyer, H.-G., and Schwefel, H.-P., 2002, "Evolution strategies: A comprehensive introduction", Natural Computing 1(1), 3–52.

Büche, D. , Stoll, P., Dornberger, R., and Koumoutsakos, P., 2002, "Multi-objective evolutionary algorithm for the optimization of noisy combustion problems, IEEE Transactions on Systems, Man and Cybernetics, Part C, 32(4), 460–473.

Büche, D., Schraudolph, N. N. , and Koumoutsakos , P. , 2005 ,"Accelerating evolutionary algorithms with gaussian process fitness function models", IEEE Transactions on Systems, Man and Cybernetics, Part C, 35(2), 183–194.

Box, G. E. P., 1957, "Evolutionary Operation: A method for increasing industrial productivity", Applied Statistics, 6(2), 81–101.

Box, G. E. P. and K. B. Wilson, 1951, "On the Experimental Attainment of Optimum Conditions", Journal of the Royal Statistical Society. Series B (Methodological), 13(1), 1–45.

Cottet, G.-H., Sbalzarini, I. F., Mueller, S. D. , and Koumoutsakos, P., 2000, “Optimization of trailing vortex destruction by evolution strategies”, *CTR Proceedings of the 2000 Summer Program*.

Ekeberg, O., 1993, “The combined neuronal and mechanical model of fish swimming”, Biological Cybernetics 69, 363–374.

Fonseca, M. C., and Fleming, P. , 1995, “Multi-objective genetic algorithms made easy: Selection, sharing and mating restrictions”, *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, No. 414, pp. 45–52.

Hansen, N., and Ostermeier, A., 1996, “Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation”, In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 312-317.

Hansen, N., and Ostermeier, A., 2001, “Completely derandomized selfadaptation in evolution strategies”, Evol. Comput. 9(2), 159–195.

Hansen, N., Müller, S. D., and Koumoutsakos, P., 2003, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)” Evol. Comput. 11(1), 1–18.

Hansen, N., and Kern, S., 2004, “Evaluating the CMA Evolution Strategy on Multimodal Test Functions”, *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature PPSN VIII*, LNCS 3242, pp. 282–291.

Hansen N., Niederberger A.S.P, Guzzella L. and Koumoutsakos P., 2007, Evolutionary Optimization under Uncertainty : Application to Feedback Controllers for Thermoacoustic Instabilities, IEEE Trans. Evolut. Comput., (submitted).

Ijspeert, A., and Kodjabachia, J., 1999, “Evolution and development of a central pattern generator for the swimming of a lamprey”, Artificial Life 5 (3), 247–269.

Kern, S., Hansen, N., and Koumoutsakos, P., 2006, “Local Meta-models for Optimization Using Evolution Strategies”, Lecture Notes in Computer Science, 4193, 939–948.

Kern, S., and Koumoutsakos, P., 2006, “Simulations of optimized anguilliform swimming”, J. Exp. Biol., 209, 4841–4857.

Kwon, O. K., and Pletcher, R. H., 1981, ”Prediction of the Incompressible Flow Over a Rearward-Facing Step”, Technical Report HTL-26, CFD-4, Iowa State Univ., Ames, IA.

Lee, Y., Korpela, S. A., and Horne, R. N., 1982, ”Structure of Multi-Cellular Natural Convection in a Tall Vertical Annulus,” *Proceedings, 7th International Heat Transfer Conference*, U. Grigul et al., ed., Hemisphere Publishing Corp., Washington, D.C., Vol. 2, pp. 221–226.

Liu, H., and Kawachi, K., 1999, “A numerical study of undulatory swimming”, J. Comput. Physics 150, 223–247

Marsden, A. L., Wang, M., and Koumoutsakos P., 2002, Optimal aeroacoustic shape design using approximation modeling, Annual Research Briefs, Center for Turbulence Research. 2002, 201–213.

Milano, M., and Koumoutsakos, P., 2002, “A clustering genetic algorithm for cylinder drag optimization”, J. Comput. Phys., 175(1), 79–107.

Müller, S. D., Mezic, I., Walther, J. H., and Koumoutsakos, P., 2004, “Transverse momentum micromixer optimization with evolution strategies”, Computers & Fluids, 33(4), 521–531.

Müller, S. D., Marchetto, J. , Airaghi, S., and Koumoutsakos, P., 2002, “Optimization based on bacterial chemotaxis”, IEEE T. Evolut. Comput., 6(1), 16–29.

Niederberger, A.S.P., Schuermans, B.B.H., and Guzzella, L., 2005, “Modeling and active control of thermoacoustic instabilities”, in *16th IFAC World Congress*, Prague.

Poncet, P., Cottet, G.-H., and Koumoutsakos, P., 2005, “Control of three-dimensional wakes using evolution strategies”, C. R. Acad. Sci. Paris, Mcanique, 333, 65–77.

Poncet, P., and Koumoutsakos, P., 2005, “Optimization of vortex shedding in 3-D wakes using belt actuators”, Int. J. Offshore and Polar Engn., 15(1) .

Ramamurti, R., Sandberg, W. C., Lohner, R., Walker, J. A., and Westneat, M. W., 2002, “Fluid dynamics of flapping aquatic flight in the bird wrasse: three-dimensional unsteady computations with fin deformation”, J Exp Biol, 205, 2997–3008

Rechenberg I., 1994, Evolutionsstrategies’94. *Frommann-Holzboog*, Stuttgart-Bad Cannstatt

Runarsson, T.P., 2004, “Constrained evolutionary optimization by approximate ranking and surrogate models”, in *Parallel Problem Solving from Nature - PPSN VIII*, LNCS 3242, 401–410.

Sbalzarini, I. F. , Müller, S. D. , Koumoutsakos, P., and Cottet, G.-H., 2001, “Evolution Strategies for Computational and Experimental Fluid Dynamics Applications”, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, Morgan Kaufmann Publishers, San Francisco, CA.

Schwefel H.-P., 1977, “Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie”, Interdisciplinary systems research; 26. Birkhuser, Basel.

Skogestad, S., Postlethwaite, I., 1996, “Multivariable Feedback Control: Analysis and Design”, John Wiley and Sons Ltd., Chichester, New York.

Sparrow, E. M., 1980, “Fluid-to-Fluid Conjugate Heat Transfer for a Vertical Pipe - Internal Forced Convection and External Natural Convection”, *ASME Journal of Heat Transfer*, Vol. 102, 402–407.

Sparrow, E. M., 1980, “Forced-Convection Heat Transfer in a Duct Having Spanwise-Periodic Rectangular Protuberances”, *Numerical Heat Transfer*, Vol. 3, 149–167.

Tung, C. Y., 1982, ”Evaporative Heat Transfer in the Contact Line of a Mixture”, Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY.

Van Veldhuizen, D. A. , and Lamont, G. B. , 2000, On measuring multiobjective evolutionary algorithm performance, *Proceedings of the 2000 Congress on Evolutionary Computation*, pp. 204-211.

Zhu, Q., Wolfgang, M. J., Yue, D. K. P., and Triantafyllou, M. S., 2002, “Three-dimensional flow structures and vorticity control in fish-like swimming”, J. Fluid Mech. 468, 1–28.

Zitzler, E., and Thiele, L., 1999, Multiobjective evolutionary algorithms: A com- parative case study and the Strength Pareto Approach, IEEE Transactions on Evolutionary Computation, pp. 257–271, Vol.3, No. 4.