

On the Combined Effect of Design-space Dimensionality Reduction and Optimization Methods on Shape Optimization Efficiency

Danny D'Agostino*

*Dept. of Computer, Control, and Management Engineering "A. Ruberti", Sapienza University of Rome
Via Ariosto, 25, 00185 Rome, Italy*

Andrea Serani[†] and Matteo Diez[‡]

*CNR-INM, National Research Council-Institute of Marine Engineering
Via di Vallerano 139, 00128 Rome, Italy*

The curse of dimensionality represents a relevant issue in simulation-based shape optimization, especially when complex physics and high-fidelity computationally-expensive solvers are involved in the process and a global optimum is sought after. In order to have a deeper insight into this problem and indicate possible remedies, the present paper studies the effects of both design-space dimensionality reduction (DR) and optimization methods on the shape optimization efficiency. Linear and non-linear DR methods are used for the design-space DR, based on principal component analysis and deep autoencoders. Global and hybrid global/local deterministic derivative-free optimization algorithms (Deterministic Particle Swarm Optimization, DIviding RECTangles, Dolphin Pod Optimization, LSDFPSO, and DIRMIN-2) are applied to the original and the reduced-dimensionality design-spaces, investigating their efficiency and effectiveness. Example application is shown for the shape optimization of a destroyer-type vessel sailing in calm water at fixed speed.

I. Introduction

THE simulation-based design (SBD) paradigm has demonstrated its capability of supporting the design decision process, providing large sets of design options and reducing time and costs of the design process. The recent development of high performance computing (HPC) systems has driven the SBD towards its integration with optimization algorithms, moving the SBD paradigm further, to automatic SBD optimization (SBDO). In shape optimization, SBDO consists of three main elements: (i) a simulation tool, (ii) an optimization algorithm, and (iii) a shape modification tool, which need to be integrated efficiently and robustly (see Fig. 1, right box).

Objectives and constraint functions, in SBDO for engineering applications, are usually provided by systems of partial differential equations, often solved by black-box tools. These functions are likely affected by residuals, therefore noisy, and their derivatives are usually not directly provided. Often, the existence in the design space of local minima cannot be excluded a priori. This motivates the use of derivative-free optimization algorithm in SBDO.

Derivative-free global optimization algorithms have been developed, providing a global approximate solution to the design problem. Derivative-free global algorithms, such as DIviding RECTangle (DIRECT) [1], Particle Swarm Optimization (PSO) [2], and Dolphin Pod Optimization (DPO) [3, 4] have emerged as powerful methods to tackle complex optimization problems in several engineering fields. The robustness and versatility of these methods have allowed for their successful application to design optimization [5, 6]. The use of global techniques with CPU-time expensive solvers (for fluid dynamics, structures, etc.) implies an expensive optimization process (considering CPU-time requirement) and its efficiency remain an algorithmic and technological challenge. Although global optimization methods are a good compromise between design space exploration and exploitation, they could be trapped in local minima and the convergence to a global minimum is generally difficult to be proved. If the research region to explore is known a priori, derivative-free local optimization approaches, can give an accurate approximation of the local minimum. Nevertheless, their convergence may be computationally expensive, and the information about the research region is

*PhD Student, Sapienza University of Rome and CNR-INM, danny.dagostino@uniroma1.it

[†]Postdoctoral Research Fellow, CNR-INM, andrea.serani@insean.cnr.it

[‡]Research Scientist, CNR-INM, matteo.diez@cnr.it

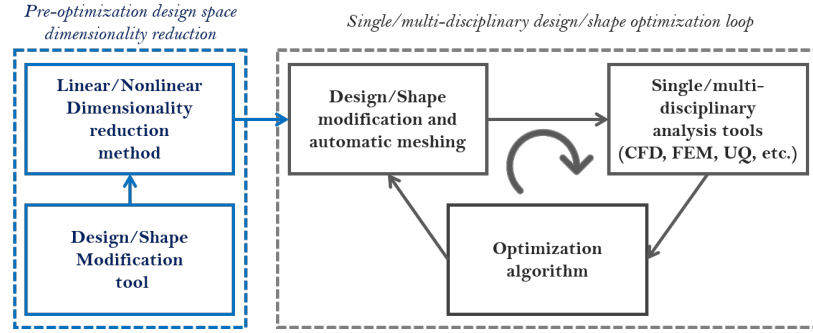


Fig. 1 SBDO scheme, including pre-optimization (offline) design-space dimensionality reduction.

usually not available a priori. For these reasons, hybrid global/local methods have been successfully applied in this context [7], especially if CPU-time expensive black-box functions are involved, coupling efficiently and robustly the qualities of both methods. It is worth noting that a large variety of derivative-free global and local methods available in the literature are probabilistic, making use of some random coefficients. This implies that statistically significant results can be obtained only through extensive numerical campaigns, that could be too expensive (often almost unaffordable) in SBDO for industrial applications, when CPU-time expensive computer simulations are used directly as analysis tools. For this reason, the use of deterministic methods, successfully developed and applied to SBDO [7, 8], is advisable.

In this context, the ability of the optimizer to fully interrogate the design space depends, among other factors, by the number of degrees of freedom (design variables) used by the shape parametrization scheme to represent a desired design within the design space. This is a critical aspect of any optimization scheme. Moreover, the use of a low number of design variables in theory is highly advantageous, particularly if a global optimization algorithm is used where the “curse of dimensionality” is particularly significant, meaning that performance of a global search algorithm deteriorates with increasing dimensions. It is reasonable that the assessment and breakdown of the design-space dimensionality and variability (see Fig. 1, left box) are therefore a key element for the success of the SBDO [9].

Online linear dimensionality reduction (DR) techniques have been developed, requiring the evaluation of the objective function or its gradient [10, 11]. As an example, principal component analysis (PCA) methods have been applied for reduced-dimensionality local representations of feasible design regions [12]. A PCA-based approach is used in the active subspace method [13] to discover and exploit low-dimensional and monotonic trends in the objective function, based on the evaluation of its gradient. Online methods improve the shape optimization efficiency by basis rotation and/or dimensionality reduction. Nevertheless, they do not provide an assessment of the design space and the associated shape parametrization before optimization is performed or objective function and/or gradient are evaluated. Offline linear DR methodologies have been developed for design optimization, with a focus on design-space variability and DR for efficient analysis and optimization procedures [14, 15] in order to deal with the curse of dimensionality. A method based on the generalized Karhunen-Loève expansion (KLE) has been formulated for the assessment of the shape modification variability and the definition of a reduced-dimensionality global model of the design space by Diez et al. [9]. No objective function evaluation nor gradient is required by the method. The method has been successfully applied for deterministic [7, 16, 17] and stochastic [18, 19] hull form optimization of mono-hulls and catamarans in calm water and waves, respectively. Similarly, Poole et al. [20] have applied proper orthogonal decomposition (POD, equivalent to generalized KLE), via singular value decomposition of an airfoils library, for airfoil shape optimization. Although linear methods have been successfully applied for a wide range of problems, they may not be efficient when complex nonlinear relationship between design variables are involved. Nonlinear DR methods generalize linear methods to address data with non-linear structures. Autoencoders are effective method for data compression; it is a particular architecture of artificial neural networks, capable to compress (encode) and reconstruct (decode) the original data. This method, in the form of deep architecture (called deep autoencoder) [21], is capable to learn complex hidden relationships about the data and to be particularly effective in presence of high nonlinearities. A deep autoencoder (DAE) has been successfully applied for pre-optimization design space breakdown of mono-hull vessel [22]. D’Agostino et al. [23] have shown the effectiveness of linear and nonlinear DR methods for the shape optimization of the destroyer type hull, with comparison of objective function improvement and convergence to the optimum.

The objective of the present paper is to study the combined effect of design-space DR and optimization methods on the efficiency of global SBDO procedures, extending the authors previous work [22, 23] where only DR methods effects

were studied.

PCA and DAE are used for the design-space assessment and breakdown of the hull-form optimization of a USS Arleigh Burke-class destroyer, namely the DTMB 5415 model, an early and open to public version of the DDG-51. The optimization pertains to the total resistance minimization in calm water at single speed performed with global (Deterministic PSO [6], DIRECT, and DPO), and global/local hybrid deterministic derivative-free methods (LSDFPSO [24] and DIRMIN-2 [5]). PCA and DAE compression capabilities are assessed in terms of original data reconstruction error by the normalized mean squared error. Their effects on the global optimization performance (using the reduced-dimensionality spaces), conditional to optimization method, is finally discussed and compared to the original full-dimensionality design space.

II. Design-space Dimensionality Reduction Methods

General definitions and assumptions for design-space DR in shape optimization are presented in the following, along with linear (PCA) and non-linear (DAE) DR methods.

A. General Definitions and Assumptions

Consider a geometric domain \mathcal{G} (which identifies the initial or parent shape) and a set of coordinates $\xi \in \mathcal{G} \subset \mathbb{R}^n$ with $n = 1, 2, 3$. Assume that $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^M$ is the design-variable vector, which defines a continuous shape modification vector $\delta(\xi, \mathbf{u}) \in \mathbb{R}^m$ with $m = 1, 2, 3$ (with m not necessarily equal to n). Consider \mathbf{u} as a random variable, with associated probability density function $p(\mathbf{u})$.

In this application the aim of the dimensionality reduction is to identify an approximated representation of the shape modification vector namely $\hat{\delta}(\xi, \mathbf{x})$, for which its shape modification depends on a new reduced order design variable $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^N$ with $N < M$. The reconstruction vector $\hat{\delta}(\xi, \mathbf{x})$ is estimated during a process of encoding/decoding by the DR methods: the original shape modification vector $\delta(\xi, \mathbf{u})$ is encoded in a low dimensional latent space defining the new design variable \mathbf{x} . The decoding process reconstruct the original shape modification vector as $\hat{\delta}(\xi, \mathbf{x})$ for which its modification on the original domain depends on the vector \mathbf{x} learned by the DR methods. Figure 2 shows an example with $n = 1$ and $m = 2$.

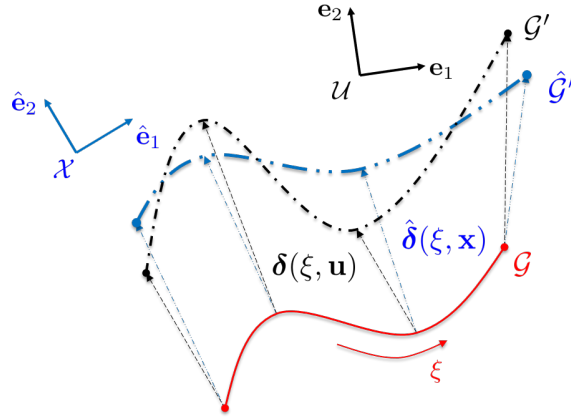


Fig. 2 Scheme and notation for the current formulation.

A convenient metric to evaluate the goodness of fit of $\hat{\delta}(\xi, \mathbf{x})$ respect to $\delta(\xi, \mathbf{u})$ is the mean squared error (MSE) normalized to the total original geometric variance (σ^2) as

$$\text{NMSE} = \frac{\text{MSE}}{\sigma^2} = \frac{\iint_{\mathcal{U} \times \mathcal{X}, \mathcal{G}} \|\delta(\xi, \mathbf{u}) - \hat{\delta}(\xi, \mathbf{x})\|^2 p(\mathbf{u}, \mathbf{x}) d\xi d\mathbf{u} d\mathbf{x}}{\iint_{\mathcal{U}, \mathcal{G}} \|\delta(\xi, \mathbf{u})\|^2 p(\mathbf{u}) d\xi d\mathbf{u}} \quad (1)$$

where $p(\mathbf{u}, \mathbf{x})$ is an unknown joint probability distribution over the product space $\mathcal{U} \times \mathcal{X}$.

Discretizing \mathcal{G} by a number of Q elements of equal measure $\Delta\mathcal{G}$, assuming for the sake of simplicity and without loss of generality $\Delta\mathcal{G} = 1$, and sampling \mathcal{U} by a statistically convergent number of Monte Carlo realizations S , so that

$\{\mathbf{u}_k\}_{k=1}^S \sim p(\mathbf{u})$. The spatial discretization $\mathbf{d}(\mathbf{u})_k$ of $\delta(\xi, \mathbf{u}_k)$ are organized in a $[S \times L]$ data matrix as:

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}(\mathbf{u})_{k=1}^\top \\ \vdots \\ \mathbf{d}(\mathbf{u})_{k=S}^\top \end{bmatrix} \quad (2)$$

where $L = mQ$ and assuming a centered data set (i.e., with zero mean value). The expectation in Eq. 1 can be approximated by evaluating the MSE as

$$\text{MSE} = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}(\mathbf{u})_k - \hat{\mathbf{d}}(\mathbf{x})_k\|^2 \quad (3)$$

and the total geometric variance σ^2 as

$$\sigma^2 = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}(\mathbf{u})_k\|^2 \quad (4)$$

Combining Eqs. 3 and 4, the discretized form of Eq. 1 can be expressed as

$$\text{NMSE} = \frac{\text{MSE}}{\sigma^2} = \frac{\sum_{k=1}^S \|\mathbf{d}(\mathbf{u})_k - \hat{\mathbf{d}}(\mathbf{x})_k\|^2}{\sum_{k=1}^S \|\mathbf{d}(\mathbf{u})_k\|^2} \quad (5)$$

Details of formulation, equations, and numerical discretization can be found in [9]. In the next sections DR methods as PCA and DAE are presented; in order to simplify notations $\mathbf{d}(\mathbf{u})$ and $\hat{\mathbf{d}}(\mathbf{x})$ are recast into \mathbf{d} and $\hat{\mathbf{d}}$, respectively.

B. Principal Component Analysis

PCA allows to reduce the input dimensionality of the data, performing a projection of the points in a new linear subspace, defined by the eigenvectors (called the principal components) of the $[L \times L]$ covariance matrix $\mathbf{C} = \frac{1}{S} \mathbf{D}^\top \mathbf{D}$. These eigenvectors have the properties to maximize the variance of points projected on them and to minimize the mean squared distance between the original points and the relative projections [25]. The principal components are defined by the solution of the eigenproblem

$$\mathbf{C} \mathbf{z}_i = \lambda_i \mathbf{z}_i \quad \forall i = 1, \dots, L \quad (6)$$

Moreover, the eigenvalues $\{\lambda_i\}_{i=1}^L$ (with $\lambda_i \geq \lambda_{i+1}$) represent the variance resolved along the relative eigenvectors $\{\mathbf{z}_i\}_{i=1}^L$ (with $\mathbf{z}_i^\top \mathbf{z}_i = 1$). From this property, a subset of N eigenvectors is used to compute a reduced dimensionality representation of the original vector \mathbf{d} as

$$\mathbf{x}_k = \mathbf{Z}^\top \mathbf{d}_k \quad (7)$$

the matrix \mathbf{Z} has dimension $[L \times N]$ and is composed by the first N largest-variance principal components. The projection $\hat{\mathbf{d}}_k$ on the orthonormal basis given by the columns of \mathbf{Z} is:

$$\hat{\mathbf{d}}_k = \mathbf{Z} \mathbf{x}_k \quad (8)$$

where $\hat{\mathbf{d}}_k$ represents the minimum squared error approximation of the relative \mathbf{d}_k :

$$\text{MSE} = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}_k - \hat{\mathbf{d}}_k\|^2 = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}_k\|^2 - \sum_{i=1}^N \mathbf{z}_i^\top \mathbf{C} \mathbf{z}_i = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}_k\|^2 - \sum_{i=1}^N \lambda_i \quad (9)$$

where in Eq. 9 it follows that in order to minimize the MSE, the principal components given by the solution of Eq. 6 also maximize the projected variance $\sum_{i=1}^N \mathbf{z}_i^\top \mathbf{C} \mathbf{z}_i$.

C. Deep Autoencoder

An autoencoder is an artificial neural network that performs two main tasks: (i) an encoder function \mathcal{E} maps the data \mathbf{d}_k to compress data \mathbf{x}_k ; (ii) a decoder function \mathcal{D} maps from the compressed data \mathbf{x}_k back to $\hat{\mathbf{d}}_k$. This operation is performed setting the same number of neurons L in the input and output layer and constraining the hidden layer to have $N < M$ neurons.

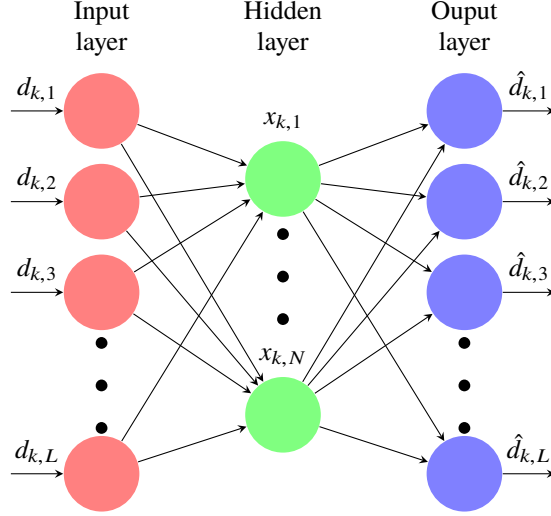


Fig. 3 A single hidden layer autoencoder.

Consider a single hidden layer autoencoder (see Fig. 3), if the new design variable \mathbf{x}_k can be written as

$$\mathbf{x}_k = \mathcal{E}(\mathbf{H}^{(1)}\mathbf{d}_k + \mathbf{b}^{(1)}) \quad (10)$$

where \mathbf{H} is a relative weight matrix, \mathbf{b} the bias vector, and the apex “(1)” represents the weights from the input to the hidden layer, then the reconstruction vector $\hat{\mathbf{d}}_k$ from \mathbf{x}_k can be expressed as

$$\hat{\mathbf{d}}_k = \mathcal{D}(\mathbf{H}^{(2)}\mathbf{x}_k + \mathbf{b}^{(2)}) \quad (11)$$

where the apex “(2)” represents the weights from the hidden to the output layer. The network parameters \mathbf{H} and \mathbf{b} , usually are evaluated minimizing the mean squared error as

$$\text{MSE} = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}_k - \hat{\mathbf{d}}_k\|^2 = \frac{1}{S} \sum_{k=1}^S \|\mathbf{d}_k - \mathcal{D}(\mathbf{H}^{(2)}\mathcal{E}(\mathbf{H}^{(1)}\mathbf{d}_k + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})\|^2 \quad (12)$$

If a linear activation function is used for the encoder function \mathcal{E} and the decoder function \mathcal{D} , the weights of the matrix $\mathbf{H}^{(1)}$ span the same subspace as the first N principal components of the data at the global minimum of Eq. 12 [26, 27]. Even if \mathcal{E} is sigmoidal activation function, the points in the hidden layer are still embedded in a hyperplane spanned by the principal components of the data [27]. Using nonlinear activation functions and multiple hidden layers, DAE provides a nonlinear generalization of the PCA. The DAE compression capability is represented by the number of neurons N in the central hidden layer and defined based on parametric minimization of the MSE, varying N .

III. Optimization Algorithms

Global and hybrid global/local deterministic derivative-free optimization algorithms are used. Their main features are described in the following subsections.

A. Deterministic Particle Swarm Optimization

PSO was originally introduced in [2], based on the social-behavior metaphor of a flock of birds or a swarm of bees searching for food. PSO belongs to the class of heuristic algorithms for single-objective evolutionary derivative-free

global optimization. In order to make PSO more efficient for use within SBD, a deterministic version of the algorithm (DPSO) was formulated in [28] as follows

$$\begin{cases} \mathbf{v}_j^{k+1} = \chi \left[\mathbf{v}_j^k + c_1(\mathbf{p}_j - \mathbf{x}_j^k) + c_2(\mathbf{g} - \mathbf{x}_j^k) \right] \\ \mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1} \end{cases} \quad (13)$$

The above equations update velocity (\mathbf{v}_j^k) and position (\mathbf{x}_j^k) of the j -th particle at the k -th iteration, where: χ is the constriction factor; c_1 and c_2 are respectively the social and cognitive learning rate; \mathbf{p}_j is the personal best position ever found by the j -th particle in the previous iterations and \mathbf{g} is the global best position ever found in the previous iterations by all the particles.

A discussion for an effective and efficient use of DPSO for SBDO in ship hydrodynamics has been presented by Serani and Diez [8]. The parameter setup used for DPSO is selected as suggested by Serani et al. [6]: number of particles $N_p = 4N$; particle initialization with Hammersley sequence sampling (HSS) [29] distribution on domain only with non-null velocity [16]; set of coefficients proposed by Clerc [30], i.e., $\chi = 0.721$, $c_1 = c_2 = 1.655$; semi-elastic wall-type approach [6] for box constraints.

B. Hybrid Global/Local DPSO

Global convergence properties of a modified DPSO scheme may be obtained by properly combining DPSO with a line search-based derivative-free method, so that convergence to stationary points can be forced at a reasonable cost. Serani et al. [24] provides a robust method to force the convergence of a subsequence of points toward a stationary point, which satisfies first order optimality conditions for the objective function. The method, namely LSDFPSO, starts by coupling the DPSO scheme with a line search-based method. Specifically, a Positively Spanning Set (PSS) is used, where the set of search directions (D_{\oplus}) is defined by the unit vectors $\pm e_i$, $i = 1, \dots, N$, as shown in the Eq. 14 (i.e., $N = 2$).

$$D_{\oplus} = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} \quad (14)$$

After each DPSO iteration, the local search from the best particle is performed if the swarm has not find a new global minimum. The initial step size for the local search is set equal to 0.25 times the variable domain range, and it is reduced by 0.5 at each local search iteration. Local searches continue in each direction until the step size is greater than 10^{-3} . If the local search stops without providing a new global minimum, the actual global minimum is declared as a stationary point. The line search method is not allowed to violate the box constraints.

C. Dividing RECTangles

DIRECT is a sampling deterministic global derivative-free optimization algorithm and a modification of the Lipschitzian optimization method [1]. It starts the optimization by transforming the search domain \mathcal{X} or \mathcal{U} of the problem into the unit hyper-cube \mathcal{D} . At the first step of DIRECT, the objective $f(\mathbf{x})$ is evaluated at the center (c) of \mathcal{D} ; the hyper-cube is then partitioned into a set of smaller hyper-rectangles and $f(\mathbf{x})$ is evaluated at their centers. Let the partition of \mathcal{D} at iteration k be defined as

$$\begin{aligned} \mathcal{H}^k &= \{\mathcal{D}^j : j \in \mathcal{J}^k\}, \text{ with} \\ \mathcal{D}^j &= \{\mathbf{x} \in \mathbb{R}^N : \mathbf{x}_l^j \leq \mathbf{x} \leq \mathbf{x}_u^j, \forall j \in \mathcal{J}^k\} \end{aligned} \quad (15)$$

where \mathbf{x}_l^j and $\mathbf{x}_u^j \in [0, 1]$, with $j \in \mathcal{J}^k$, are the lower and upper bounds defining the hyper-rectangle \mathcal{D}^j , and \mathcal{J}^k is the set of index identifying the subsets defining the current partition. At a generic k -th iteration of the algorithm, starting from the current partition \mathcal{H}^k of \mathcal{D} , a new partition, \mathcal{H}^{k+1} , is built by subdividing a set of promising hyper-rectangles of the previous one. The identification of ‘‘potentially optimal’’ hyper-rectangles is based on a trade-off between large and unexplored hyper-rectangles and the value of $f(\mathbf{x})$ at their center c^j . The refinement of the partition continues until a prescribed number of function evaluations have been performed, or another stopping criterion is satisfied. The minimum of $f(\mathbf{x})$ over all the centers of the final partition, and the corresponding center, provide an approximate solution to the problem. It may be noted that the box constraints are automatically satisfied.

D. Hybrid Global/Local DIRECT

DIRMIN-2 is a global/local hybridization of the DIRECT algorithm [5]. DIRMIN-2 performs a single derivative-free local minimization per iteration, starting from the best point produced by dividing the potentially-optimized hyper-rectangles. DIRMIN-2's local minimization is used when the number of function evaluations reaches the activation trigger $\gamma \in (0, 1)$, a ratio of the maximum number of function evaluations ($N_{f_{\max}}$). The local minimization proceeds until either the number of function evaluations exceeds $N_{f_{\max}}$ or the step size falls below a given tolerance β . The local search is not allowed to violate the box constraints. The following set up is used [5] $\gamma = 0$ and $\beta = 10^{-2}$.

E. Dolphin Pod Optimization

DPO is a deterministic swarm-intelligence method based on a simplified social model of a dolphin pod in search for food, originally introduced by Serani and Diez [3, 4]. DPO is formulated defining the pod dynamics as a spring-mass system subject to internal and external forces. DPO formulation starts normalizing the variables vector \mathbf{x} into a unit hypercube and reads

$$\begin{cases} \mathbf{v}_j^{k+1} = (1 - \zeta \frac{\Delta t}{p}) \mathbf{v}_j^k - \frac{\Delta t}{p N_d} (\alpha_1 \boldsymbol{\psi}_j - \alpha_2 \boldsymbol{\varphi}_j) \\ \mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1} \frac{\Delta t}{p} \end{cases} \quad (16)$$

where \mathbf{x}_j^k and \mathbf{v}_j^k represent the j -th dolphin position and velocity at the k -th iteration, respectively. The dynamics of the j -th individual depends on a pod attraction force $\boldsymbol{\psi}_j$ and a food attraction force $\boldsymbol{\varphi}_j$, where

$$\boldsymbol{\psi}_j = \sum_{i=1}^{N_d} (\mathbf{x}_j - \mathbf{x}_i) \quad \text{and} \quad \boldsymbol{\varphi}_j = \sum_{i=1}^{N_d} \frac{2 \hat{f}(\mathbf{x}_j, \mathbf{b}_i)}{1 + \|\mathbf{x}_j - \mathbf{b}_i\|^{\alpha_3}} \mathbf{e}(\mathbf{b}_i, \mathbf{x}_j) \quad (17)$$

with

$$\hat{f}(\mathbf{x}_j, \mathbf{b}_i) = \frac{f(\mathbf{x}_j) - f(\mathbf{b}_i)}{\rho} \quad \text{and} \quad \mathbf{e} = \frac{\mathbf{b}_i - \mathbf{x}_j}{\|\mathbf{b}_i - \mathbf{x}_j\|} \quad (18)$$

In the above equations, ζ , α_1 , and $\alpha_2 \in \mathbb{R}^+$ define the pod dynamics; $N_d \in \mathbb{N}^+$ is the pod size; $\alpha_3 \in \mathbb{R}^+$ tunes the food attraction force; $\mathbf{x}_j \in \mathbb{R}^N$ is the vector-valued position of the j -th individual; $f(\mathbf{x}) \in \mathbb{R}$ is the objective function (representing the food distribution); \mathbf{b}_i is the best position ever visited by the i -th individual; $\rho = f(\mathbf{w}) - f(\mathbf{b})$ is a dynamic normalization term for f , where $\mathbf{b} = \operatorname{argmin}\{f(\mathbf{b}_j)\}$ is the best position ever visited by the pod and $\mathbf{w} = \operatorname{argmax}\{f(\mathbf{x}_j)\}$ the worst position occupied by the pod individuals at the current time instance; \mathbf{b}_i , \mathbf{b} , and \mathbf{w} are defined in the variable space. The integration step Δt must guarantee the stability of the pod free dynamics as per

$$\Delta t = \frac{2\gamma}{\gamma^2 + \omega^2} \Big|_{\min} \quad (19)$$

where γ and ω are real and imaginary part, respectively, of the matrix \mathbf{A} eigenvalue $\lambda = -\gamma \pm i\omega$, describing the system free-dynamics, with

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{G} \end{bmatrix}, \quad \mathbf{K} = \alpha_1 \begin{bmatrix} 1 - N_d & 1 & \cdots & 1 \\ 1 & 1 - N_d & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & 1 - N_d \end{bmatrix}, \quad \text{and} \quad \mathbf{G} = \zeta \mathbf{I} \quad (20)$$

where \mathbf{I} the $[N_d \times N_d]$ identity matrix.

The setup suggested by Serani and Diez [4] is used herein: number of dolphins $N_d = 4N$; pod initialization with HSS distribution on the domain boundary only; set of coefficients corresponding to $\zeta = 0.1$, $\alpha_1 = \alpha_2 = 0.1$, $\alpha_3 = 0.5$, and $p = 8$; elastic wall-type approach for box constraints.

IV. Shape Optimization of a Destroyer Hull

A. Model Geometry and Optimization Problem Statement

The DTMB 5415 model is an open-to-public early concept of the DDG-51, a USS Arleigh Burke-class destroyer, widely used for both towing tank experiments [31] and hull-form SBDO [7]. The original (full dimensionality) design

space formulation for the shape optimization of the DTMB 5415 reads

$$\begin{aligned}
& \text{minimize} && R_T(\mathbf{u}) && \text{with } \mathbf{u} \in \mathbb{R}^M \\
& \text{subject to} && L_{pp}(\mathbf{u}) = L_{pp0} \\
& \text{and to} && \nabla(\mathbf{u}) = \nabla_0, \\
& && |\Delta B(\mathbf{u})| \leq 0.05B_0, \\
& && |\Delta T(\mathbf{u})| \leq 0.05T_0, \\
& && V(\mathbf{u}) \geq V_0, \\
& && u_i^l \leq u_i \leq u_i^u \quad \forall i = 1, \dots, M
\end{aligned} \tag{21}$$

where R_T is the calm-water total resistance at $Fr = 0.28$ (equivalent to 20 kn for the full-scale ship). Equality constraints are defined for the length between perpendiculars (L_{pp}) and for the displacement (∇). Inequality constraints include 5% of maximum variation of beam (B) and the draught (T) and preserved volume for the sonar in the bow dome V . Subscript ‘0’ indicates original-geometry values.

Using the reduced-dimensionality space dimension N , the optimization problem in Eq. 21 is recast as

$$\begin{aligned}
& \text{minimize} && R_T(\mathbf{x}) && \text{with } \mathbf{x} \in \mathbb{R}^N \\
& \text{subject to} && L_{pp}(\mathbf{x}) = L_{pp0} \\
& \text{and to} && \nabla(\mathbf{x}) = \nabla_0, \\
& && |\Delta B(\mathbf{x})| \leq 0.05B_0, \\
& && |\Delta T(\mathbf{x})| \leq 0.05T_0, \\
& && V(\mathbf{x}) \geq V_0, \\
& && x_i^l \leq x_i \leq x_i^u \quad \forall i = 1, \dots, N
\end{aligned} \tag{22}$$

where the objective function and constraints depend on the new N -dimensional design variable vector \mathbf{x} estimated by the off-line design space dimensionality reduction (Fig. 1); x_i^l and x_i^u are the lower and the upper bounds respectively computed from the minimum and the maximum of each column of the $[S \times N]$ matrix, composed by all the encoded representation of $\{\mathbf{d}_k\}_{k=1}^S$ (see Eq. 7 and Eq. 10).

B. Shape Modification Method

The original shape modification $\delta(\boldsymbol{\xi}, \mathbf{u})$ is defined using a linear combination of $M = 27$ vector-valued functions of the Cartesian coordinates $\boldsymbol{\xi} \in \mathbb{R}^3$ over a hyper-rectangle embedding the demihull [7]

$$\boldsymbol{\phi}_i(\boldsymbol{\xi}) : \mathcal{V} = [0, L_{\xi_1}] \times [0, L_{\xi_2}] \times [0, L_{\xi_3}] \in \mathbb{R}^3 \longrightarrow \mathbb{R}^3 \tag{23}$$

with $i = 1, \dots, M$, as

$$\delta(\boldsymbol{\xi}, \mathbf{u}) = \sum_{i=1}^M u_i \boldsymbol{\phi}_i(\boldsymbol{\xi}) \tag{24}$$

where the coefficients $u_i \in \mathbb{R}$ ($i = 1, \dots, M$) are the (original) design variables and

$$\boldsymbol{\phi}_i(\boldsymbol{\xi}) := \prod_{j=1}^3 \sin\left(\frac{a_{ij}\pi\xi_j}{L_{\xi_j}} + r_{ij}\right) \mathbf{e}_{q(i)} \tag{25}$$

imposing the following orthogonality property:

$$\int_{\mathcal{V}} \boldsymbol{\phi}_i(\boldsymbol{\xi}) \cdot \boldsymbol{\phi}_k(\boldsymbol{\xi}) d\boldsymbol{\xi} = \delta_{ik} \tag{26}$$

In Eq. 25, $\{a_{ij}\}_{j=1}^3 \in \mathbb{R}$ define the order of the function along j -th axis; $\{r_{ij}\}_{j=1}^3 \in \mathbb{R}$ are the corresponding spatial phases; $\{L_{\xi_j}\}_{j=1}^3$ are the hyper-rectangle edge lengths; $\mathbf{e}_{q(i)}$ is a unit vector. Modifications are applied along ξ_1 , ξ_2 , or ξ_3 , with $q(i) = 1, 2$, or 3 respectively. The parameter values used here are taken from [7].

In the discrete form, the original shape modification reads

$$\mathbf{g}(\mathbf{u}) = \mathbf{g}_0 + \mathbf{d}(\mathbf{u}) \quad (27)$$

where \mathbf{g} is the computational body grid (used for the hydrodynamics analysis, see Fig. 4), and \mathbf{g}_0 is the original grid.

During the data matrix \mathbf{D} construction, the equality and inequality constraints of problem in Eqs. 21 and 22 are taken in consideration: fixed L_{pp} and ∇ are satisfied by automatic geometric scaling, whereas geometries exceeding the inequality constraints are not considered. Once the design-space dimensionality reduction is performed, the shape modification (Eq. 27) in the reduced-dimensionality space is given by

$$\mathbf{g}(\mathbf{x}) = \mathbf{g}_0 + \hat{\mathbf{d}}(\mathbf{x}) \quad (28)$$

with $\hat{\mathbf{d}}(\mathbf{x})$ equal to Eqs. 8 and 11 for PCA and DAE, respectively.

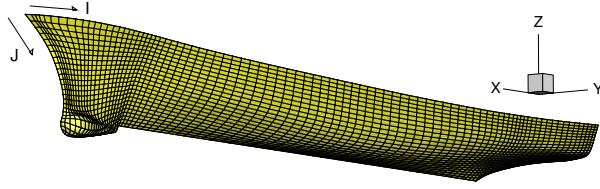


Fig. 4 DTMB 5415 geometry and body surface discretization used for both design-space dimensionality reduction and optimization.

C. Hydrodynamic Solver

Calm-water performance are evaluated using the linear potential-flow code WARP (Wave Resistance Program), developed at CNR-INSEAN. Wave resistance computations are based on the on Dawson (double-model) linearization [32]. The wave resistance is evaluated with the pressure integral over the body surface, whereas the frictional resistance is estimated using a flat-plate approximation, based on the local Reynolds number [33]. The ship balance (sinkage and trim) is fixed. Details of equations, numerical implementations, and validation of the numerical solver are given in Bassanini et al. [34].

Simulations are performed for the right demihull, taking advantage of symmetry about the xz -plane. The computational domain for the free-surface is defined within $1L_{pp}$ upstream, $3L_{pp}$ downstream, and $1.5L_{pp}$ sideways, for a total of 75×20 grid nodes. The associated hull grid (90×25 nodes) is shown in Fig. 4.

V. Results

Design-space dimensionality reduction by PCA and DAE and the optimization results using both full and reduced-dimensionality design spaces are presented in the following subsections.

A. Design-space Dimensionality Reduction

The original design space (defined by $M = 27$ design variables) is sampled using a uniform random distribution of $S = 9,000$ hull-form designs. In assessing the methods' performance and to address the overfitting problem the DR methods are trained on the 80% of the original dataset \mathbf{D} (training set). The remaining 20% is used as test set to evaluate DR effectiveness in terms of reconstruction error. In the present work, DAE is built with five hidden layer composed by 160-50- N -50-160 neurons. Exponential linear units (ELUs) [35] activation function is chosen for each hidden layers, except for the output layer where there is a linear activation function. The DAE is trained with Adam optimization algorithm [36] using a minibatch size of 64 data point when compute the gradient by the backpropagation algorithm [37]. To overcome the DAE overfitting the early stopping strategy is adopted during the training: a subset of the training set, namely the validation set, is used to evaluate the NMSE during the learning process, which stops if no NMSE improvement is found. A maximum threshold of the NMSE equal to 0.05 is set to identify the dimension N of the reduced-dimensionality spaces. This value describes data variability that DR methods are not capable to explain.

Figure 5 compares PCA and DAE normalized reconstruction error convergence (on the test set) towards the NMSE threshold. PCA needs $N = 21$ principal components to reach the threshold, reducing the design-space dimensionality up to 22%, whereas DAE outperform the PCA reaching about 56% with $N = 12$. This is an indicator of high nonlinearity in the data, since the PCA overestimates the intrinsic dimensionality of the data.

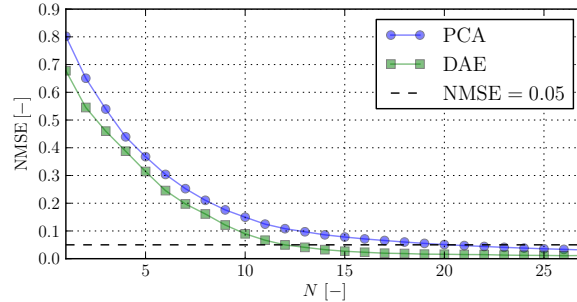


Fig. 5 PCA and DAE normalized reconstruction error convergence on the test set.

B. Optimization

A limit to the number of objective function evaluations is set equal to 9,000. A linear penalty function is used for the designs exceeding the inequality constraints.

Figure 6 shows the optimization algorithms convergence towards the optimum using original (OBF, see Fig. 6a), PCA (see Fig. 6b), and DAE (see Fig. 6c) spaces. A maximum total resistance improvement of about 13%, 16%, and 25% is achieved by DPSO, DIRECT, and DIRMIN-2, respectively on OBF, PCA, and DAE spaces. This first result shows how the DR methods generally improve the global optimization procedure.

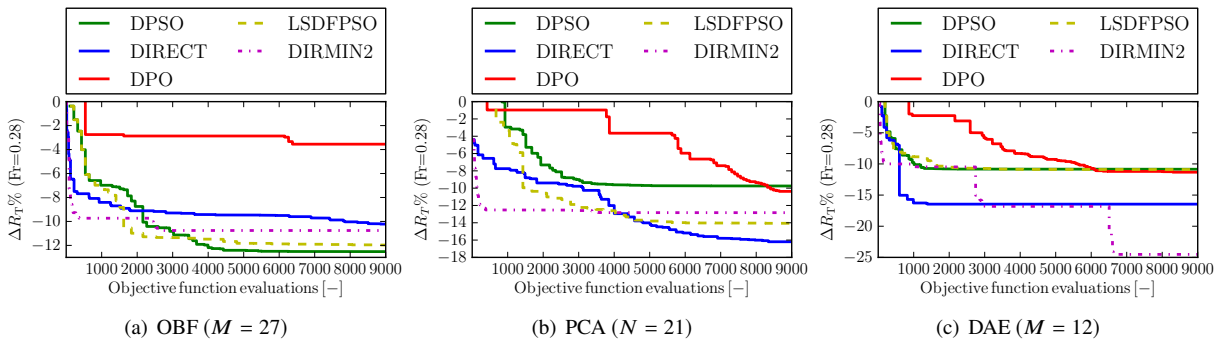


Fig. 6 Optimization algorithms convergence conditional to the design space.

To further assess the effects of the DR methods on the optimization procedure, Fig. 7 shows for each optimization algorithms the objective function converge, conditional to the design space. DPSO (see Fig. 7a) and its hybrid counterpart LSDLFPSO (see Fig. 7c) don't seem to take any benefit from using PCA and DAE design spaces. Nevertheless about 3% of maximum difference between the optima for both DPSO and LSDLFPSO has been found. On the contrary, DIRECT (see Fig. 7b) and its hybrid DIRMIN-2 (see Fig. 7d), as well as DPO (see Fig. 7c), significantly increase their performance in terms of convergence speed and objective function improvement as the problem dimensionality decrease.

Finally, Fig. 8 and Tab. 1 summarize the optimization results in terms of total resistance improvement conditional to the optimization algorithm and the design space. Average objective improvement conditional to the design space is also provided, showing how the dimensionality reduction and in particular a nonlinear methods, such as the DAE, generally improves the optimization effectiveness and efficiency.

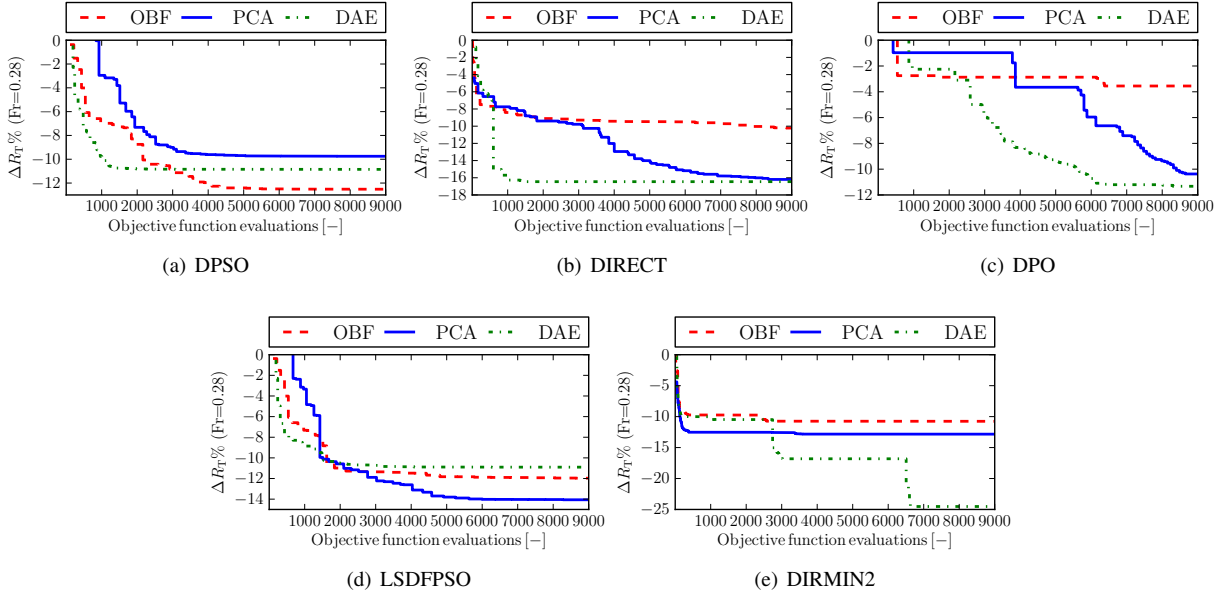


Fig. 7 Design space effectiveness conditional to the optimization algorithm.

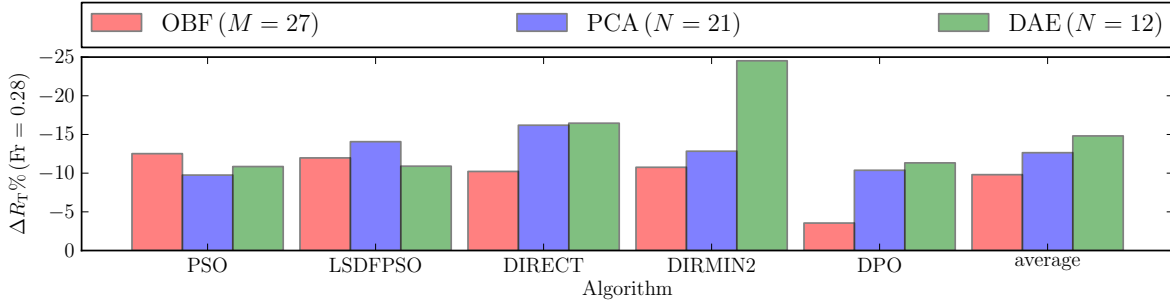


Fig. 8 Optimization improvement using 9,000 function evaluations conditional to design spaces and algorithm. Algorithms average is also provided.

Table 1 Optimization results summary: objective function improvement ($\Delta R_T\%$).

Design space	Dimension	DPSO	LSDFPSO	DIRECT	DIRMIN-2	DPO	Average
OBF	$M = 27$	-12.51	-11.96	-10.22	-10.75	-3.55	-9.80
PCA	$N = 21$	-9.75	-14.06	-16.19	-12.83	-10.37	-12.64
DAE	$N = 12$	-10.85	-10.89	-16.45	-24.52	-11.32	-14.81

VI. Conclusions and Future Work

The combined effect of linear and nonlinear design-space DR and optimization methods on global shape optimization efficiency has been investigated, providing a deeper insight into the curse of dimensionality for complex simulation-based problems. Linear PCA and nonlinear DAE are used as DR methods. Deterministic Particle Swarm Optimization, Diving Rectangles, and hybrid global/local counterpart (LSDFPSO and DIRMIN-2), as well as Dolphin Pod Optimization, are used as optimization algorithms. Example application is shown for the shape optimization of the DTMB 5415 model, a destroyer type vessel, addressing the calm water total resistance minimization at fixed speed.

DR methods have shown their capability of reducing the original design-space dimensionality ($M = 27$) by up to 22% (PCA) and 56% (DAE), achieving a reconstruction error based on the NMSE smaller than 5%. As far as the DR

effectiveness is concerned, DAE clearly outperforms PCA.

From the optimization efficiency point of view, Lipschitzian optimization algorithms (DIRECT and its hybrid modification DIRMIN-2) and DPO, a nature inspired algorithm, benefited the most from using reduced-dimensionality spaces both in terms of convergence speed to the minimum and final objective function improvement. No significant benefits of using reduced-dimensionality spaces have been observed for DPSO and its hybrid version, LSDFPSO. On average, a total resistance improvement of 9.8%, 12.6%, and 14.8% has been achieved using original (OBF), PCA, and DAE design spaces, respectively, providing a clear indication of the general benefit of using reduced-dimensionality spaces.

Future work will extend the current study to additional nonlinear DR methods such as kernel and local PCA [22]. Application of nonlinear methods will be also extended and assessed for combined geometry- and physics-based formulation for design-space DR [38–40].

Acknowledgments

The work is partially supported by the US Department of the Navy Office of Naval Research Global, NICOP grant N62909-15-1-2016 and N62909-18-1-2033, administered by Dr. Woei-Min Lin, Dr. Salahuddin Ahmed, and Dr. Ki-Han Kim, and by the Italian Flagship Project RITMARE, funded by the Italian Ministry of Education. The research is performed within NATO STO Task Group AVT-252 “Stochastic Design Optimization for Naval and Aero Military Vehicles”.

References

- [1] Jones, D., Perttunen, C., and Stuckman, B., “Lipschitzian optimization without the Lipschitz constant,” *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, 1993, pp. 157–181.
- [2] Kennedy, J., and Eberhart, R., “Particle swarm optimization,” *Proceedings of the Fourth IEEE Conference on Neural Networks*, Piscataway, NJ, 1995, pp. 1942–1948.
- [3] Serani, A., and Diez, M., “Dolphin Pod Optimization,” *Advances in Swarm Intelligence. ICSI 2017*, edited by Y. Tan, H. Takagi, and Y. Shi, Lecture Notes in Computer Science, vol 10385, Springer, Cham, 2017, pp. 63–70.
- [4] Serani, A., and Diez, M., “Dolphin Pod Optimization: A Nature-Inspired Deterministic Algorithm for Simulation-Based Design,” *3rd International Conference on Machine Learning, Optimization, and Big Data, MOD 2017, Volterra, Italy*, 2017.
- [5] Campana, E. F., Diez, M., Iemma, U., Liuzzi, G., Lucidi, S., Rinaldi, F., and Serani, A., “Derivative-free global ship design optimization using global/local hybridization of the DIRECT algorithm,” *Optimization and Engineering*, Vol. 17, No. 1, 2015, pp. 127–156.
- [6] Serani, A., Leotardi, C., Iemma, U., Campana, E. F., Fasano, G., and Diez, M., “Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems,” *Applied Soft Computing*, Vol. 49, 2016, pp. 313 – 334.
- [7] Serani, A., Fasano, G., Liuzzi, G., Lucidi, S., Iemma, U., Campana, E. F., Stern, F., and Diez, M., “Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms,” *Applied Ocean Research*, Vol. 59, 2016, pp. 115 – 128.
- [8] Serani, A., and Diez, M., “Are Random Coefficients Needed in Particle Swarm Optimization for Simulation-Based Ship Design?” *Proceedings of the 7th International Conference on Computational Methods in Marine Engineering (Marine 2017)*, 2017.
- [9] Diez, M., Campana, E. F., and Stern, F., “Design-space dimensionality reduction in shape optimization by Karhunen–Loève expansion,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 283, 2015, pp. 1525–1544.
- [10] Berguin, S. H., and Mavris, D. N., “Dimensionality Reduction Using Principal Component Analysis Applied to the Gradient,” *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 1078 – 1090.
- [11] Berguin, S. H., Rancourt, D., and Mavris, D. N., “Method to Facilitate High-Dimensional Design Space Exploration Using Computationally Expensive Analyses,” *AIAA Journal*, Vol. 53, No. 12, 2015, pp. 3752 – 3765.
- [12] Raghavan, B., Breikopf, P., Tourbier, Y., and Villon, P., “Towards a space reduction approach for efficient structural shape optimization,” *Structural and Multidisciplinary Optimization*, Vol. 48, 2013, p. 987–1000.

- [13] Lukaczyk, T., Palacios, F., Alonso, J. J., and Constantine, P., "Active subspaces for shape optimization," *Proceedings of the 10th AIAA Multidisciplinary Design Optimization Specialist Conference, National Harbor, Maryland, USA, 13-17 January, 2014*.
- [14] Borzì, A., Schulz, V., Schillings, C., and von Winckel, G., "On the treatment of distributed uncertainties in PDE-constrained optimization," *GAMM-Mitteilungen*, Vol. 33, No. 2, 2010, pp. 230–246.
- [15] Schillings, C., Schmidt, S., and Schulz, V., "Efficient shape optimization for certain and uncertain aerodynamic design," *Computers & Fluids*, Vol. 46, No. 1, 2011, pp. 78–87.
- [16] Chen, X., Diez, M., Kandasamy, M., Zhang, Z., Campana, E. F., and Stern, F., "High-fidelity global optimization of shape design by dimensionality reduction, metamodels and deterministic particle swarm," *Engineering Optimization*, Vol. 47, No. 4, 2015, pp. 473–494.
- [17] Diez, M., Serani, A., Campana, E. F., Volpi, S., and Stern, F., "Design Space Dimensionality Reduction for Single- and Multi-Disciplinary Shape Optimization," *AIAA/ISSMO Multidisciplinary Analysis and Optimization (MA&O), AVIATION 2016*, Washington D.C., USA, June 13-17, 2016.
- [18] Diez, M., Campana, E. F., and Stern, F., "Development and evaluation of hull-form stochastic optimization methods for resistance and operability," *Proceedings of the 13th International Conference on Fast Sea Transportation, FAST 2015*, Washington, D.C., USA, 2015.
- [19] Diez, M., Campana, E. F., and Stern, F., "Stochastic optimization methods for ship resistance and operational efficiency via CFD," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 2, 2018, pp. 735–758.
- [20] Poole, D., Allen, C., and Rendall, T., "High-fidelity aerodynamic shape optimization using efficient orthogonal modal design variables with a constrained global optimizer," *Computers & Fluids*, Vol. 143, No. Supplement C, 2017, pp. 1 – 15.
- [21] Hinton, G. E., and Salakhutdinov, R. R., "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, No. 5786, 2006, pp. 504–507.
- [22] D'Agostino, D., Serani, A., Campana, E. F., and Diez, M., "Nonlinear Methods for Design-Space Dimensionality Reduction in Shape Optimization," *3rd International Conference on Machine Learning, Optimization, and Big Data, MOD 2017, Volterra, Italy*, 2017.
- [23] D'Agostino, D., Serani, A., Campana, E. F., and Diez, M., "Deep Autoencoder for Off-line Design-Space Dimensionality Reduction in Shape Optimization," *56th AIAA Aerospace Sciences Meeting, SciTech 2018*, Gaylord Palms, Kissimmee, Florida, USA, January 8-12, 2018.
- [24] Serani, A., Diez, M., Campana, E. F., Fasano, G., Peri, D., and Iemma, U., "Globally Convergent Hybridization of Particle Swarm Optimization Using Line Search-Based Derivative-Free Techniques," *Recent Advances in Swarm Intelligence and Evolutionary Computation*, Studies in Computational Intelligence, Vol. 585, edited by X.-S. Yang, Springer International Publishing, 2015, pp. 25–47.
- [25] Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [26] Baldi, P., and Hornik, K., "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, Vol. 2, No. 1, 1989, pp. 53–58.
- [27] Bourlard, H., and Kamp, Y., "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, Vol. 59, No. 4, 1988, pp. 291–294.
- [28] Campana, E. F., Liuzzi, G., Lucidi, S., Peri, D., Piccialli, V., and Pinto, A., "New global optimization methods for ship design problems," *Optimization and Engineering*, Vol. 10, No. 4, 2009, pp. 533–555.
- [29] Wong, T. T., Luk, W. S., and Heng, P. A., "Sampling with Hammersley and Halton Points," *Journal of Graphics Tools*, Vol. 2, No. 2, 1997, pp. 9–24.
- [30] Clerc, M., "Stagnation analysis in particle swarm optimization or what happens when nothing happens," Tech. rep., 2006. URL <http://hal.archives-ouvertes.fr/hal-00122031>.
- [31] Stern, F., Longo, J., Penna, R., Olivieri, A., Ratcliffe, T., and Coleman, H., "International Collaboration on Benchmark CFD Validation Data for Surface Combatant DTMB Model 5415," *Proceedings of the Twenty-Third Symposium on Naval Hydrodynamics*, Val de Reuil, France, September 17-22, 2000.

- [32] Dawson, C. W., “A practical computer method for solving ship-wave problems,” *Proceedings of the 2nd International Conference on Numerical Ship Hydrodynamics*, Berkeley, 1977, pp. 30–38.
- [33] Schlichting, H., and Gersten, K., *Boundary-Layer Theory*, Springer-Verlag, Berlin, 2000.
- [34] Bassanini, P., Bulgarelli, U., Campana, E. F., and Lalli, F., “The wave resistance problem in a boundary integral formulation,” *Surveys on Mathematics for Industry*, Vol. 4, 1994, pp. 151–194.
- [35] Clevert, D.-A., Unterthiner, T., and Hochreiter, S., “Fast and accurate deep network learning by exponential linear units (ELUs),” *arXiv preprint arXiv:1511.07289*.
- [36] Kingma, D., and Ba, J., “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*.
- [37] Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al., “Learning representations by back-propagating errors,” *Cognitive Modeling*, Vol. 5, No. 3, 1988, p. 1.
- [38] Diez, M., Serani, A., Stern, F., and Campana, E. F., “Combined Geometry and Physics Based Method for Design-Space Dimensionality Reduction in Hydrodynamic Shape Optimization,” *Proceedings of the 31st Symposium on Naval Hydrodynamics, Monterey, CA, USA*, 2016.
- [39] Serani, A., Campana, E. F., Diez, M., and Stern, F., “Towards Augmented Design-Space Exploration via Combined Geometry and Physics Based Karhunen-Loève Expansion,” *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MA&O), AVIATION 2017*, Denver, USA, June 5-9, 2017.
- [40] Serani, A., and Diez, M., “Shape Optimization under Stochastic Conditions by Design-space Augmented Dimensionality Reduction,” *19th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (MA&O), AVIATION 2018*, Atlanta, USA, June 25-29, 2018.