International Journal for Numerical Methods in Engineering

WILEY

RESEARCH ARTICLE

Data-Driven Parameterization Refinement for the Structural Optimization of Cruise Ship Hulls

Lorenzo Fabris[1] | Marco Tezzele[2] | Ciro Busiello[3] | Mauro Sicchiero[3] | Gianluigi Rozza[1]

[1]Mathematics Area, mathLab, SISSA, Scuola Internazionale Superiore di Studi Avanzati, Trieste, Italy | [2]Department of Mathematics, Emory University, Atlanta, Georgia, USA | [3]Merchant Ships Business Unit, Fincantieri S.p.A., Trieste, Italy

Correspondence: Gianluigi Rozza (gianluigi.rozza@sissa.it)

Received: 17 June 2025 | **Revised:** 9 October 2025 | **Accepted:** 14 November 2025

Keywords: Bayesian optimization | multi-objective optimization | reduced order modeling | structural optimization

ABSTRACT

In this work, we focus on the early design phase of cruise ship hulls, where the designers are tasked with ensuring the structural resilience of the ship against extreme waves while reducing steel usage and respecting safety and manufacturing constraints. At this stage, the geometry of the ship is already finalized, and the designer can choose the thickness of the primary structural elements, such as decks, bulkheads, and the shell. Reduced order modeling and black-box optimization techniques reduce the use of expensive finite element analysis (FEA) to only validate the most promising configurations, thanks to the efficient exploration of the domain of decision variables. However, the quality of the final results heavily relies on the problem formulation and on how the structural elements are assigned to the decision variables. A parameterization that does not capture well the stress configuration of the model prevents the optimization procedure from achieving the most efficient allocation of the steel. With the increased request for alternative fuels and engine technologies, the designers are often faced with unfamiliar structural behaviors and risk producing ill-suited parameterizations. To address this issue, we enhanced a structural optimization pipeline for cruise ships developed in collaboration with Fincantieri S.p.A. with a novel data-driven hierarchical reparameterization procedure, based on the optimization of a series of subproblems. Moreover, we implemented a multi-objective optimization module to provide the designers with insights into the efficient trade-offs between competing quantities of interest and enhanced the single-objective Bayesian optimization (BO) module. The new pipeline is tested on a simplified midship section and a full ship hull, comparing the automated reparameterization to a baseline model provided by the designers. The tests show that the iterative refinement outperforms the baseline on the more complex hull, proving that the pipeline streamlines the initial design phase and helps the designers tackle more innovative projects. The reparameterization procedure only relies on the evaluation of surrogate models and can be applied with minimal changes to other large-scale structural problems where yielding and buckling constitute the limiting factor to the design.

1 | Introduction

The shipbuilding industry faces continuously evolving requirements, as the rise in environmental consciousness prescribes reducing operational costs and adopting new engine technologies [1]. The hull design should reflect a lower resource usage during the manufacturing and operational phase, while at the same time accommodating new machinery such as liquid hydrogen tanks or batteries. All of these innovations must then observe the structural stability constraints imposed by the classification societies, which guarantee the safety and durability of the ship. The implementation of flexible and efficient optimization procedures during the initial design phase thus represents a crucial factor for innovation and competitiveness. However, the formulation of the optimization problem is a delicate task. When facing a project with novel characteristics, the designers might not anticipate the emergence of fringe behaviors and corner cases for which their initial formulation produces underwhelming results. In this work, we minimize the total mass of a cruise ship during the initial design phase, combining surrogates-assisted optimization and an automatic refining strategy for the optimization problem. The validation of the optimized designs is carried out through finite element analysis (FEA), integrating the tools and pipelines familiar to the designers.

FEA is widespread in industry [2], and commercial solvers are available for the certification of industrial designs. However, the computational cost of FEA is incompatible with rapid iterations of optimization steps in the initial design phase of large-scale projects. Advanced FEA codes provide adjoint-based optimizers [3], but the implementation of complex and sometimes intractable rules from the classification societies hinders their adoption into the design workflow. Recent works on the structural optimization of marine artifacts have been focused on the use of either simplified analytical formulations or FEA of small, incomplete models. A multi-objective genetic algorithm was used to reduce mass and structural instability of a ship, with the values of objectives and constraints computed from the FEA of a single hull ring [4].

Finite differences of the simplified expressions for yielding and buckling were used to find search directions for the optimization of a semisubmersible floater [5]. FEA of a coarse mesh, combined with a simplified analytical model for the stiffener structures, was optimized with particle swarm optimization [6]. A collection of recent publications on design methods for the marine industry, including structural and topology optimization, can be found in the report of the International Ship and Offshore Structures Congress [7].

Reduced order models (ROMs) [8–10] provide an alternative to FEA in the initial design phase, enabling the designers to quickly iterate changes to parameter values and shape configurations, with the expensive validations being performed only on the most promising configurations. Nonintrusive approaches [11, 12] are able to separate the optimization procedure from the underlying physical model, a crucial requirement for workflows relying on closed-source commercial codes. Still, the optimization results are heavily dependent on the problem formulation: previous approaches often constrained entire decks and bulkheads to use a single thickness value, so that local insurgence of high stress could not be prevented without great waste of mass. On the other hand, excessively fine parameterizations[1] result in an extremely large domain space, which is difficult to explore efficiently.

In this work, we present an automated reparameterization procedure in the context of structural optimization, where the problem formulation is refined hierarchically through an integer linear problem (ILP) [13] based on the structural responses observed near the best-known configuration. The procedure extends an existing automated optimization pipeline for the optimization of passenger ship hulls [14], developed in collaboration with Fincantieri S.p.A., where data-driven surrogates based on proper orthogonal decomposition (POD) and Gaussian process regression (GPR) [15, 16] are optimized using Bayesian optimization (BO) [17]. Moreover, the framework is extended with the addition of multi-objective optimization through a genetic algorithm (GA) [18–21], and the integration of constraints for the vertical center of gravity (VCG) in the BO procedure. Single-objective optimization is further refined with a specialization for the discrete parametric domain, and a greedy heuristic that we call principal dimension search (PDS). Figure 1 shows the end-to-end pipeline, from the initial parameterization to the optimal hull. The process begins with an initial formulation of the optimization problem and the high-fidelity evaluation of a randomized sampling of the parametric domain. The high-fidelity evaluations are used to build surrogates for the cheap evaluation of the quantities of interest (QoIs). The following optimization steps, multi and single-objective, use the surrogates to efficiently select promising parameter configurations to expand the high-fidelity evaluations and refine the surrogates. Then, an automatic refinement of the problem formulation is performed, increasing the capability of reaching better values of the QoIs at the cost of introducing more decision variables. Following the problem refinement, the sequence of surrogate construction and optimization is started again, until the user is satisfied with the QoIs of the high-fidelity optimum or the stopping criteria are met.

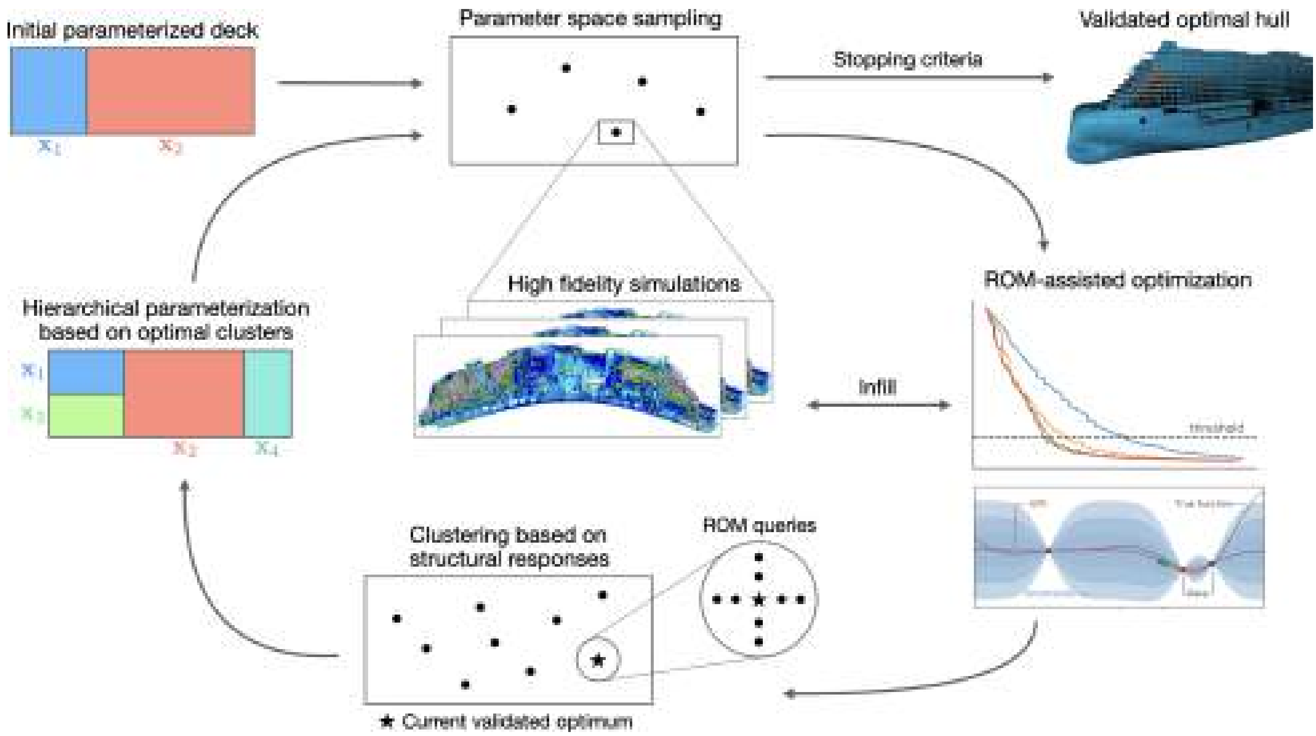

FIGURE 1 | Optimization pipeline implementing the inner-outer loop approach. The process starts with a coarse parameterization of the model and a random sampling of the parametric space. The inner loop consists of the high-fidelity, expensive simulation of the samples, from which ROMs are built in the outer loop. The ROMs are optimized with a multi-objective GA and single-objective BO, with infill criteria selecting the most promising configurations for addition to the high-fidelity database. The reparameterization procedure starts when the optimization step is unable to find new candidates. The ROMs are queried in the proximity of the current optimum to collect the structural responses, which are then clustered to generate a finer parameterization of the model, producing a new set of decision variables adapted to the emerging structural behavior. The new decision variables are hierarchically dependent on the previous ones, and the high-fidelity database is easily updated. New samples from the larger parametric domain are drawn, and the cycle repeats until the stopping criteria are met.

The paper is organized as follows. The problem formulation for the initial design phase is described in Section 2, along with the proposed solution for an automatic optimization pipeline. In Section 3, we discuss the numerical methods for the implementation of the optimization pipeline. The pipeline is applied to two test cases, a simplified midship section and a full ship model, with the results being analyzed in Section 4. Finally, we draw conclusions in Section 5.

2 | Problem Definition

The design phase of cruise ships can take more than one year to be completed, and is subdivided into multiple subphases which deal with increasing levels of detail. In this work, we concentrate on the early design phase, where the designers are tasked with ensuring the structural resilience of the ship against extreme waves, while reducing steel usage and respecting safety and manufacturing constraints. At this point in the project, the design of decks, shells, cabins, and functional areas has been finalized at a coarse level, defining the 2D geometry of the steel plates that constitute the primary structural components of the ship. To modify the structural response of the model, the designers can act on the thickness and the steel type used in the primary structural components, or they can change the configuration of the secondary structural components (such as beams and flanges) by varying their size and spacing.

Every configuration needs to be validated through a computationally expensive FEA simulation. Combined with the large variability in the design of cruise ship, which often include ad-hoc elements such as theaters, elevators disposition, or open areas, the initial design phase usually lasts several months. In this work, we implement an optimization framework to speed up the design process and provide the designers with optimal configurations.

In Section 2.1, we present the FEA setup and the QoIs extracted by a simulation. In Section 2.2 we present the optimization pipeline.

2.1 | Structural Problem Formulation

The global model of a ship hull uses quadrilateral and triangular shell elements for the primary structural members (decks, bulkheads, and shell), while the secondary stiffeners also use beam elements. The shell elements have principal dimensions of approximately 700 mm. The load conditions applied to the mesh come from two extreme waves, as defined by the classification society: the hogging condition corresponds to the wave's crest being placed at the ship's mid-length, while in the sagging condition, the ship's middle point is placed above the trough. Figure 2 shows the two load conditions applied to the model of a full ship. The model loads are then completed by the machinery and furniture. From a global structural strength perspective, ships are designed to withstand the maximum expected loads throughout their service life, typically 20 years. In accordance with the requirements of all major classification societies, the resulting stresses must remain below the material yield limit, with an appropriate safety factor applied to ensure structural integrity. Consequently, linear static models are deemed sufficient for structural analyses, as prescribed by classification standards. For cruise ships, the most critical conditions for global structural assessment are those of maximum hogging and maximum sagging. Other load conditions, such as racking (where the waves impact the ship from the sides), typically affect localized structures (e.g., large transverse bulkheads) whose structural sizing has a limited impact on overall weight optimization.

MSC NASTRAN [22] assembles a linear static analysis problem for a given parameter configuration, that is, a parameterized hull, using thin plate theory for the constitutive equations of shell elements. The outputs of interest of the simulation comprise the three-dimensional displacements of the mesh nodes and the in-plane stress tensors evaluated at the centroids of the shells' faces. The stress tensors of each shell face are averaged and transformed to the global coordinate reference system for further analysis. At this point, the stress tensors are collected as

$$S_e^l = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_y & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_z, \end{bmatrix}, \tag{1}$$

where l denotes the load condition, e the shell element index, σ the direct stresses, and τ the shear components. Subscripts to the stress components identify the axis, or plane, of action.

To quantify the structural integrity of the ship under the given load conditions, the stress tensor fields are post-processed to compute the yielding and buckling states of each element, according to the rules imposed by the classification society. In this paper, the structural criteria come from the Det Norske Veritas (DNV) specifications for ships of a length of 100 meters and above. The yielding state is obtained by comparing the stress tensor components with the appropriate allowable value and evaluating the von Mises yielding criterion, that is,

$$|\sigma_i| \leq 245 \text{ N/mm}^2, \qquad \text{for } i \in \{x, y, z\}, \tag{2a}$$

$$|\tau_i| \leq 153 \text{ N/mm}^2, \qquad \text{for } i \in \{xy, xz, yz\}, \tag{2b}$$

$$\sigma_{\mathrm{VM}} = \sqrt{\frac{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2}{2} + 3(\tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2)} \leq 307 \text{ N/mm}^2, \tag{2c}$$

where the critical values refer to high-strength structural steel (AH36). For an element to not yield, all conditions in Equation (2) must hold for all load conditions; otherwise, the element is marked as yielded.

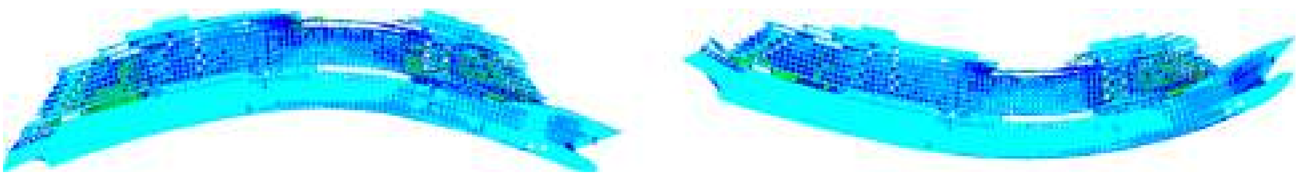

FIGURE 2 | A full ship model under hogging load condition on the left, and sagging on the right. Displacements are magnified, colors represent the value of the von Mises yielding criterion.

Although the used shell elements do not model out-of-plane stresses, the DNV specifies buckling criteria [23] computed from the stress tensor components and the geometry of the panel the element is part of. These criteria are based on general elastic buckling formulae, corrected in the plastic range, and result in 11 usage factors. An element is marked as buckled if any of these usage factors is above a certain threshold for at least one load condition.

Both yielding and buckling phenomena can be corrected with additional manufacturing actions, albeit at the cost of increased build time and resource usage. These corrections are undesirable, and the designers specify critical thresholds on the number of yielded and buckled elements, above which a design is to be penalized. These thresholds provide the main constraints of the initial design phase, and the total number of yielded and buckled elements constitutes the first QoIs obtained from a high-fidelity simulation. The mass and cost of the model are computed from the volume of the steel used and the amount of adopted corrective measures. In this work, we only consider buckling correction through the application of secondary stiffeners, which affect both total mass and cost.

The vertical deflection is computed as the maximum absolute vertical displacement of a selected mesh node, and is interpreted as an index of the structural stiffness of the hull. Lower vertical deflection values correspond to more resilient ships. The VCG, computed for the structural components of the ship's weight, is required to be lower than a critical value determined by the geometry of the hull.

The list of QoIs computed from a simulation is thus composed of five scalars: number of yielded elements, number of buckled elements, total mass considering the corrective actions, vertical deflection, and VCG.

The decision variables for the mass optimization problem are the thickness of selected shell elements. The domain $\mathcal{T}$ for these variables is discrete, since it is limited to commercially available sizes. A subset of the available thicknesses $\mathcal{D}_i \subseteq \mathcal{T}$ is assigned to the i-th group of elements to be controlled by the parameter $\mathbf{x}_i$, since regulations can specify different minimum thicknesses depending on the part of the ship. In this work, only primary members can be controlled by parameters, while the thickness of secondary stiffeners and the rest of the material properties are fixed. The evaluation of a single configuration requires assembling and solving the corresponding FEA problem, followed by post-processing of the results to obtain the QoIs.

2.2 | Optimization Pipeline

We build on a previous optimization pipeline adopted by Fincantieri, employing data-driven surrogates and BO to minimize the total mass of the model [14]. The inner-outer loop approach iterates two phases, the inner one being the high-fidelity simulation of one or more parameter configurations, and the outer phase being constituted by the selection of new configurations to be simulated, chosen through optimization of the surrogates. The high-fidelity simulations are computationally expensive; thus, it is assumed that only a limited number of runs are possible at each iteration. The outer loop builds data-driven surrogates based on the high-fidelity simulations, with the following optimizations using a black-box approach. The closed-source high-fidelity FEA solver used does not allow access to the implementation details, preventing intrusive approaches to model order reduction. The loop of outer and inner operations is interrupted when the outer phase is unable to further select promising candidates. The end goal of our pipeline is to help the designers quickly identify the capabilities of the model in terms of critical structural behaviors. The optimal design found by the automated procedure will necessarily undergo further analysis and modifications beyond what is modeled in our setup, but the time and resources saved in this initial phase allow the designer to focus only on the more complex problems. In this paper, we expand the existing pipeline with additional steps in the outer phase, as depicted in Figure 1. We added a multi-objective optimization module and a reparameterization module. The reparameterization is the main innovation, and it does not rely on the particular optimization algorithms chosen in the outer loop. The complete pipeline, represented as a flowchart, can be seen in Figure A1.

Multi-objective optimization provides an effective tool for the exploration of the model capabilities through the discovery of optimal trade-offs between contrasting QoIs, such as total mass and vertical deflection, enabling better-informed decisions on the constraints for the optimization problem. Even for simple designs, the trade-offs between the QoIs reveal multimodal relations between the parameters. Approaches based on random sampling of the parametric space are not able to properly identify the configurations with the optimal trade-offs, without using an overwhelming number of samples.

The reparameterization module aims to reformulate the optimization problem itself by increasing the parameters to improve the best-known configuration. Indeed, the initial parameterization of the model often proves to be a limiting factor in the efficacy of the optimization, either due to being usually based on simplified beam theory or due to the model having complex or novel structures. The reparameterization procedure identifies whether a group of elements controlled by the same parameter can be split, creating new independent parameters so that different thickness values can be assigned to the subgroups. The following optimization is then able to reduce the mass of elements that exhibit less yielding and buckling phenomena, while increasing the stiffness of highly stressed parts of the hull. Designers are no longer limited by the chosen initial parameterization, where a modification would require a tedious manual reconfiguration and data transfer.

Finally, a number of improvements to the single-objective optimization module enable a more efficient exploration of the parametric domain, which becomes necessary to handle the growing number of parameters coming from the reparameterization step.

3 | Numerical Methods

This section presents the numerical methods used in the automatic optimization pipeline. Section 3.1 discusses the implementation of data-driven ROMs to compute the QoIs, Section 3.2 illustrates the optimization procedures enabled by the surrogates, and Section 3.3 describes the parameterization refinement process.

3.1 | Model Order Reduction

To implement a computationally cheap and accurate surrogate of the stress tensor field for different load conditions, we combine POD and GPR. In this subsection, we briefly present these two methods.

3.1.1 | Proper Orthogonal Decomposition

POD decomposes a matrix of high-fidelity snapshots to obtain an orthogonal basis for the span of the column space. Let $\mathbf{M} \in \mathbb{R}^{n \times m}$ be a matrix collecting the m snapshots as columns of n features, and assume $n \gg m$, which corresponds to operating in a large-scale scenario. Then, the singular value decomposition (SVD) of $\mathbf{M}$, that is, $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, uniquely determines $\mathbf{U} \in \mathbb{R}^{n \times n}$ the matrix with the left singular vectors of $\mathbf{M}$ as columns, $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times m}$ the diagonal matrix of singular values, and $\mathbf{V}^T \in \mathbb{R}^{m \times m}$ the transpose of the matrix with the right singular vectors of $\mathbf{M}$ as columns. Uniqueness is achieved by sorting the singular values in decreasing order, up to a change of sign of the basis vectors or a change in positions corresponding to repeated singular values. By choosing a truncation rank $r < m$, one can introduce the truncated SVD

$$\mathbf{M} \approx \tilde{\mathbf{M}} = \tilde{\mathbf{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{V}}^T, \tag{3}$$

which constructs $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ by selecting the r leftmost columns of $\mathbf{U}$ and $\mathbf{V}$, and the corresponding r largest singular values form the diagonal matrix $\tilde{\boldsymbol{\Sigma}}$. The error obtained in the reconstruction of the snapshot matrix with the truncated modes can be quantified, in Frobenius norm, as

$$||\mathbf{M} - \tilde{\mathbf{M}}||_F^2 = \sum_{i=r+1}^{m} \sigma_i^2, \tag{4}$$

and $\tilde{\mathbf{M}}$ coincides with the optimal linear reconstruction. This reconstruction error is commonly referred to as “residual energy” [24]. The reduced coefficients can be computed through the projection of the snapshots matrix $\mathbf{M}$ unto the basis $\tilde{\mathbf{U}}$, with the matrix $\mathbf{C} = \tilde{\mathbf{U}}^T\mathbf{M}$ holding as i-th column the reduced coefficients of for the i-th snapshot in $\mathbf{M}$. As each snapshot is associated with a parameter configuration, the reduced coefficients provide the ground truth for the construction of a map from the parametric domain to the space of reduced coefficients. In this paper, the snapshots are obtained through the execution of closed-source commercial codes; thus, regression can only be carried out by data-driven methods. Successful approaches that have been leveraged in the past are radial basis functions interpolation, GPR and artificial neural networks, among others [25–28].

3.1.2 | Gaussian Process Regression

GPR assumes that any set of observations of the QoI is sampled from a multivariate normal distribution. This section follows the discussion from [29]. Another excellent resource is [16]. Let $\mathbf{y}$ be a vector of observations $\mathbf{y}_i = f(\mathbf{x}^{(i)})$, and $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. The vector of marginal means $\boldsymbol{\mu}$ and the covariance matrix $\mathbf{K}$ are defined as:

$$\boldsymbol{\mu}_i = \mathbb{E}\left[\mathbf{y}_i\right] = \mathbb{E}\left[f(\mathbf{x}^{(i)})\right], \tag{5}$$

$$\mathbf{K}_{ij} = \mathbf{K}_{ji} = \mathbb{E}\left[(\mathbf{y}_i - \boldsymbol{\mu}_i)(\mathbf{y}_j - \boldsymbol{\mu}_j)\right] = \mathbb{C}\mathrm{ov}\left[\mathbf{y}_i, \mathbf{y}_j\right] = \mathbb{C}\mathrm{ov}\left[f(\mathbf{x}^{(i)}), f(\mathbf{x}^{(j)})\right]. \tag{6}$$

The conditional distribution of a subset of observations L, on past outcomes H, can be obtained by leveraging the joint distribution

$$\begin{bmatrix} \mathbf{y}_H \\ \mathbf{y}_L \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_H \\ \boldsymbol{\mu}_L \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{HH} & \mathbf{K}_{HL} \\ \mathbf{K}_{LH} & \mathbf{K}_{LL} \end{bmatrix} \right), \tag{7}$$

to obtain

$$\mathbf{y}_{L|H} \sim \mathcal{N}(\boldsymbol{\mu}_{L|H}, \mathbf{K}_{L|H}), \tag{8}$$

where the conditioned terms are expressed in terms of the mean and covariance matrix of the process's distribution as

$$\boldsymbol{\mu}_{L|H} = \boldsymbol{\mu}_L + \mathbf{K}_{LH}\mathbf{K}_{HH}^{-1}(\mathbf{y}_H - \boldsymbol{\mu}_H), \tag{9}$$

$$\mathbf{K}_{L|H} = \mathbf{K}_{LL} - \mathbf{K}_{LH}\mathbf{K}_{HH}^{-1}\mathbf{K}_{HL}. \tag{10}$$

The crucial component of a GPR is the construction of two functions $\mu(\mathbf{x}^{(i)})$ and $\mathrm{kern}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ to generate the entries of $\boldsymbol{\mu}$ and $\mathbf{K}$, respectively, so that past observations are reproduced with high confidence and extrapolation beyond the observed instances gives a coherent probability distribution. The mean function, which encodes any prior knowledge about the output, is usually set to zero. Indeed, in Equation (9) the covariance-dependent terms act as a correction, weighting the observed deviations from the prior. For the covariance function, which is required to generate a symmetric semidefinite positive matrix, a popular choice is the squared exponential kernel with automatic relevance determination:

$$\mathbf{K}_{ij} = \mathrm{kern}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sigma^2 \exp\left(-\frac{1}{2} \sum_d \frac{(\mathbf{x}_d^{(i)} - \mathbf{x}_d^{(j)})^2}{l_d^2} \right), \tag{11}$$

where σ^2 is a scaling factor, and l_d is the length scale associated to the dimension d of the parametric domain.

Let $\mathbf{K}_\theta$ denote the covariance matrix obtained by the chosen kernel, whose hyperparameters are collected in the vector $\boldsymbol{\theta}$. For the squared exponential kernel, $\boldsymbol{\theta} = [\sigma, l_1, \dots, l_{n_D}]$ where n_D is the number of dimensions of the parametric domain. For the case of zero prior and the kernel function in Equation (11), with $\mathbf{X} = \left[\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\right]$, it reads

$$\log p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\frac{m}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}_\theta| - \frac{1}{2}\mathbf{y}^T\mathbf{K}_\theta^{-1}\mathbf{y}, \tag{12}$$

where $|\mathbf{K}_\theta|$ is the determinant of the covariance matrix.

Once the chosen kernel has been fitted to the available snapshots, the values in $\boldsymbol{\theta}$ are kept constant and the term $\mathbf{K}_{HH}^{-1}(\mathbf{y}_H - \boldsymbol{\mu}_H)$ in Equation (9) can be computed once and stored. Thus, the online query for the conditioned mean requires only the computation of the kernel function between the snapshots samples, and the query point. The execution time of a GPR query on a single-point scale with the product of the number of snapshot samples, output dimensions, and kernel function complexity.

3.1.3 | Implementation of the Surrogates

In this paper, our aim is to find a computationally inexpensive map from the thickness configuration to the stress tensor fields. We leverage POD to reduce the dimensionality of the output space and use GPR to map the parameter configurations to the vector of reduced coefficients, thus obtaining a much simpler problem setup. This approach is known as POD-GPR in the literature, and has been successfully used in solid mechanics [28, 30, 31]. In the following, we omit the indices for the load condition and the stress tensor component to simplify the notation.

Let $\mathbf{S} \in \mathbb{R}^{n \times m}$ be the matrix collecting the snapshots vectors $\mathbf{s}^{(i)} := \mathbf{s}(\mathbf{x}^{(i)})$ of the current stress tensor component and load condition, associated with the parameter configuration $\mathbf{x}^{(i)}$. Let $\tilde{\mathbf{U}}$ be the basis matrix obtained by the SVD of $\mathbf{S}$, truncated at rank r. We obtain the parameterized reduced coefficients through the projection

$$\mathbf{c}^{(i)} := \mathbf{c}(\mathbf{x}^{(i)}) = \tilde{\mathbf{U}}^T \mathbf{s}^{(i)}, \tag{13}$$

For each vector of r reduced coefficients, we train a different GPR on the m pairs $(\mathbf{x}^{(i)}, \mathbf{c}^{(i)})$, so that the approximated quantities, denoted by a hat, for the coefficients and the original field satisfy

$$\mathbf{s}^{(i)} \approx \hat{\mathbf{s}}^{(i)} = \tilde{\mathbf{U}} \hat{\mathbf{c}}^{(i)}, \tag{14}$$

where the hat denotes the result of an approximation.

In this paper, we consider two load conditions (hogging and sagging) and the unique six elements of the Cauchy stress tensor Equation (1); therefore, the total number of vector-valued GPRs being fitted is 12. As a remark, the actual number of stress tensor components for quad and tria elements is 3, since these are limited to two principal dimensions and a single shear. However, the industrial process that the pipeline must interact with has been developed internally, and validated externally, for many years using stress tensors in global coordinates, thus requiring six components. An alternative to our choice would be training 3 GPRs for the 3D stress tensors and applying the rotations from local to global coordinates. However, the time overhead due to accessing or assembling the matrices was paired with a negligible gain in accuracy in terms of QoIs, and this approach was discarded. Another option for the design of the surrogates is to concatenate the stress tensor fields and reduce the taller snapshots matrix with a single SVD, but this approach can quickly incur high memory requirements and a harsh trade-off when parallelizing. We use vector-valued GPRs, each mapping a parameter configuration to the r POD coefficients corresponding to a different combination of load condition and stress tensor component. We also observe that the decay of singular values consistently achieves relative residual energy (Equation (4)) below 1% for rank r higher than, but close to, the number of parameters. This behavior results in an effective model order reduction and results in a cheap training phase for the GPRs. Additionally, the construction of the GPRs for each stress tensor component and load condition is embarrassingly parallel, once the tensor fields are decomposed and each component is stored separately.

The post-processing steps for the computation of the yielding and buckling states of each element only require the stress tensor components and metadata of the element, leading to an efficient vectorized implementation. The same holds for the aggregation to compute the total number of yielded and buckled elements, from which the derivation of total mass and cost is straightforward.

We remark that the use of an efficient post-processing step, instead of the direct implementation of surrogates for the QoIs, results in a much lower error in the prediction [14]. For the surrogate of the vertical deflection, we construct a single GPR for each load condition and, during a query, take the maximum of the absolute value of the predictions.

The vector of QoIs computed from a parameter configuration $\mathbf{x}$ is thus

$$[n_{\mathrm{y}}(\mathbf{x}), n_{\mathrm{b}}(\mathbf{x}), f_{\mathrm{deflection}}(\mathbf{x}), f_{\mathrm{mass}}(\mathbf{x}), \mathrm{VCG}(\mathbf{x})] \tag{15}$$

where $n_{\mathrm{y}}(\mathbf{x})$ is the number of elements for which the predicted stress tensor, under any load condition, fails at least one of Equation (2). The number of buckled elements $n_{\mathrm{b}}(\mathbf{x})$ follows the same principle, but comes from the aggregation of the buckling usage factors computed according to [23]. $f_{\mathrm{deflection}}(\mathbf{x})$ is the vertical deflection, while the mass is computed as

$$f_{\mathrm{mass}}(\mathbf{x}) = m_{\mathrm{fixed}} + \mathbf{d} \cdot \mathbf{x} + m_{\mathrm{bar}} n_{\mathrm{b}}(\mathbf{x}), \tag{16}$$

where m_{fixed} represents the mass of the elements not controlled by any parameter and $\mathbf{d}$ is the vector collecting the linear density of each parameterized section, where the i-th component is given by the ratio of total mass of the shell elements assigned to the i-th section, divided by the thickness value. Finally, m_{bar} is the average mass of a reinforcement bar. Then, the VCG is computed as

$$\mathrm{VCG}(\mathbf{x}) = \frac{\mathrm{VCG}_{\mathrm{fixed}} m_{\mathrm{fixed}} + \sum_i \mathrm{VCG}_i \mathbf{d}_i \mathbf{x}_i}{m_{\mathrm{fixed}} + \sum_i \mathbf{d}_i \mathbf{x}_i} \geq 0, \tag{17}$$

where $\mathrm{VCG}_{\mathrm{fixed}}$ is the VCG of elements not controlled by any parameter, and VCG_i is the VCG of the i-th parameterized region. VCG is conventionally assumed to be positive.

Overall, the computational complexity of querying our surrogates' scales as $\mathcal{O}(r(n + md))$, where d is the number of parameters. With both $m \ll n$ and $d \ll n$, the cost of a surrogate query is much lower than the $\mathcal{O}(n^2)$ required by a high-fidelity FEA for a sparse linear system. This enables the use of the surrogates in many-query and quasi-real-time applications, such as black-box optimization procedures and graphical user interfaces.

3.2 | Surrogate-Assisted Optimization

The surrogates for the stress field, combining POD and GPR, allow a large number of inexpensive queries on the parametric domain. Only the most favorable configurations, selected by optimization, will be validated with an expensive full-order simulation. However, as in many industrial applications, the parametric domain is large enough that exhaustive exploration is still unfeasible, and an infill criterion is required. We propose a combination of a multi-objective GA, BO, and a heuristic local search to guide the exploration of the parametric domain and perform full order simulations only for the most promising configurations.

In Section 3.2.1, we present the GA for multi-objective optimization, and an infill criterion which leverages the uncertainty quantification provided by the GPR component of our surrogates. Section 3.2.2 presents a specialization of BO, discussing the choice of objective function and the implementation of additional constraints and heuristics. Section 3.2.3 illustrates the black-box local heuristic used to further refine the optimum found by BO.

3.2.1 | Multi-Objective Optimization

Often, multiple QoIs are used as objective functions, and it is not clear whether a global criterion for the prioritization of one over another exists. In this case, the experts are interested in obtaining not a single optimal configuration, but rather a set of configurations exhibiting the best trade-offs between the different QoIs. Such a set is called the Pareto set (PS) of the problem, and its image in the space of QoIs is the Pareto frontier or front (PF). By analyzing the PF, the experts can compromise between the various needs and impose stricter or laxer constraints for further single-objective optimization tasks.

Let $\mathcal{D}$ be a parametric domain, and f a vector of QoIs $\{f_i : \mathcal{D} \to \mathbb{R}\}_{i=1}^{n}$ which must be all minimized. A configuration $\overline{\mathbf{x}} \in \mathcal{D}$ is called dominated if there is at least one $\mathbf{x} \in \mathcal{D}$ such that $f_i(\mathbf{x}) \leq f_i(\overline{\mathbf{x}})$ for all i, and $f_i(\mathbf{x}) < f_i(\overline{\mathbf{x}})$ for at least one i. A dominated configuration is of no interest in the optimization since there exists another configuration that performs no worse on all QoIs, and is strictly better for at least one QoI. Given a population of configurations, the PS is then taken as the subset of nondominated individuals.

In our framework, we use multi-objective optimization of the QoIs shown in Equation (15) to enrich the initial sampling of high-fidelity simulations and provide the designers with an approximated PF. The designers can then analyze the PF and decide on the critical thresholds for the following single-objective optimization by estimating the heuristic trade-offs between mass and structural QoIs.

To approximate the PF, many popular approaches use GAs [18–20] in which an initial population is iteratively grown through crossover and mutation of its best individuals, and the least desirable elements are selected for removal. Other approaches from the literature include the construction of the PS by collaboration between multiple random scalarizations of the QoIs [32]. A number of methods have been developed for probabilistic surrogates, extending single-objective BO to the multi-objective setting [33–35]. These methods usually maintain multiple GPRs built from a sample population, where the addition of new samples comes by optimizing some measure of quality of the PF. The usual choice is maximizing the dominated hyper-volume, as with the expected hyper-volume improvement (EHVI) [36]. However, these methods add a further level of surrogation, require Monte Carlo evaluation of the EHVI against the current PF, and the acquisition procedure needs to be adapted to the discrete parametric domain of our problem. We chose to adopt a GA to directly query the surrogates described in Section 3.1.3, which support efficient batch evaluations. The parametric domain is explored by modifying the population according to the crossover, mutation, and selection operators, which are implemented to guarantee the feasibility and diversity of the samples. The generic structure is outlined in Algorithm 1.

ALGORITHM 1 | Template of a genetic algorithm.

```
Require: initial population $\mathbf{X}$, fitness function $f$, crossover and mutation
         functions, maximum population $p$, number of generations $q$
Ensure: final population
  $i \leftarrow 0$
  while $i<q$ do
    $\mathbf{X}^{\text{parents}} \leftarrow$ select from $\mathbf{X}$ according to $f$
    $\mathbf{X}^{\text{children}} \leftarrow$ crossover of $\mathbf{X}^{\text{parents}}$
    $\mathbf{X}^{\text{children}} \leftarrow$ mutation of $\mathbf{X}^{\text{children}}$
    $\mathbf{X} \leftarrow \mathbf{X}^{\text{parents}} \cup \mathbf{X}^{\text{children}}$
    $\mathbf{X} \leftarrow$ the $p$ best-performing individuals from $\mathbf{X}$ according to $f$
    $i \leftarrow i+1$
  end while
  return $\mathbf{X}$
```

ALGORITHM 2 | Nondominated sorting.

```
Require: list of objective functions evaluations $\mathbf{Y} = \left\{\mathbf{y}^{(i)}\right\}_{i=1}^{m}$
Ensure: list $L$ of nondominated layers as a set of indices
  $L \leftarrow$ empty list
  $R \leftarrow \{i = 1, \dots, m\}$
  while $|R|>0$ do
    $\mathbf{Y}^{\mathrm{R}} \leftarrow \{\mathbf{y}^{(i)} \mid i \in R\}$
    $M \leftarrow \{i \mid \mathbf{y}^{(i)}$ is not dominated in $\mathbf{Y}^{\mathrm{R}}\}$
    $L \leftarrow L \cup M$
    $R \leftarrow R \setminus M$
  end while
  return $L$
```

The crossover operator aims to mix the features of a set of parents, to generate one or more new members of the population. Classic approaches are single- and two-point crossover, interpolation between the parents' features, and uniform crossover [29]. In our implementation, we use two-point crossover and a variant of interpolation crossover where we take the element-wise minimum or maximum of the parents' features. The mutation operator instead introduces new traits in the population by random mutation of existing individuals. In our implementation, we choose at random whether to decrease or increase the features of an individual, or to generate a completely new one.

For the selection of fittest individuals, we use the nondominated sorting strategy proposed in the NSGA family of algorithms [21, 37]. Individuals are ranked by their degree of domination with Algorithm 2.

The selection of the fittest individuals is performed by taking the union of nondominated subsets, starting from the first, until the limit of population size is reached. If the number of selected individuals exceeds the limit, the subset with the most dominated individuals is down-sampled with the niching strategy from NSGA-III [37], to prevent overcrowding in the QoIs space.

Due to the use of surrogates, the final PF is only an approximation of the true set of nondominated individuals. An infill criterion is needed to select a reduced number of individuals to be validated by the high-fidelity solver, so that further multi-objective optimizations yield a more accurate PF. The approximated PS comprises the high-fidelity samples already validated and which contribute to the construction of the surrogates and the low-fidelity samples, which can be added to the high-fidelity set to increase the accuracy of the surrogates. For an individual to be chosen, it is required that its addition to the high-fidelity samples maximally improve the predictions at the location of the remaining low-fidelity members of the PF. This approach is similar to [38], which adapts the idea of maximizing the conditioned uncertainty reduction from [39] and the integration of an uncertainty measure over a domain of interest from [40], to the case of GPR. This principle has been developed in terms of entropy [41–43] for multi-objective optimization for GPRs that directly model the objective functions. We choose to measure the covariance between the samples in the reduced coefficient space since the corresponding GPRs have been thoroughly validated as providing accurate predictions of the stress tensors, from which the QoIs are derived. Instead of explicitly estimating the reduction in uncertainty as in previous works [38], we estimate the increment in covariance, which only requires the evaluation of the kernel and is thus cheap compared to Equation (10). Moreover, we operate on a set of GPRs so that an aggregation strategy becomes necessary.

The infill procedure starts by constructing the covariance matrices between the elements in $\mathbf{X}^{\mathrm{L}}$, and between those in $\mathbf{X}^{\mathrm{L}}$ and $\mathbf{X}^{\mathrm{H}}$. We aggregate over reduced coefficients and load conditions by taking the maximum, then sum over the stress tensor components to obtain $\mathbf{C}^{\mathrm{LL}} \in \mathbb{R}^{n\times n}$, symmetric, and $\mathbf{C}^{\mathrm{LH}} \in \mathbb{R}^{n\times m}$.

Let $\{\mathbf{K}_{\mathrm{LL}}^{(l,s,c)}\}$ be the set of covariance matrices of the GPR corresponding to the reduced coefficient c of the stress tensor component s under the load condition l, evaluated between the samples in $\mathbf{X}^{\mathrm{L}}$. Then, the element (i, j) of the aggregation matrix is computed as

$$\mathbf{C}_{ij}^{\mathrm{LL}} = \sum_{c} \max_{l,s} \left[\left(\mathbf{K}_{\mathrm{LL}}^{(l,s,c)} \right)_{ij} \right], \qquad \forall\ i, j = 1, \dots, n, \tag{18}$$

giving a scalar measure of the mutual information between the samples in $\mathbf{X}^{\mathrm{L}}$. The same expression is used for the construction of $\mathbf{C}^{\mathrm{LH}}$. A vector Δ is built by computing, for the i-th sample in $\mathbf{X}^{\mathrm{L}}$, the total positive relative increase over the maximum high-fidelity contribution to the rest of the low-fidelity PF as

$$\Delta_i = \frac{\sum_{j\neq i} (\mathbf{C}_{ij}^{\mathrm{LL}} - \max_h \mathbf{C}_{jh}^{\mathrm{LH}})_+}{\sum_{j\neq i} \max_h \mathbf{C}_{jh}^{\mathrm{LH}}}, \qquad \forall\ i = 1, \dots, n, \tag{19}$$

where $(\cdot)_+ = \max(\cdot, 0)$. The high-fidelity set is then enriched by adding the i^*-th sample determined by

$$i^* = \arg\max_i \Delta. \tag{20}$$

The choice of the aggregation functions in Equation (18) can be viewed as constructing the best-case scenario for the mean covariance's increase across the stress tensor components. If more than one configuration can be selected for the high-fidelity validation, the matrices $\mathbf{C}^{\mathrm{LL}}$ and $\mathbf{C}^{\mathrm{LH}}$ can be updated by simulating the addition of the sample i^* to the high-fidelity set, as summarized in Figure 3. In particular, the i^*-th row (and column) is removed from $\mathbf{C}^{\mathrm{LL}}$, its elements are used to build the column $m+1$ in $\mathbf{C}^{\mathrm{LH}}$, and finally the i^*-th row of $\mathbf{C}^{\mathrm{LH}}$ is removed. The resulting matrices have thus size $n-1 \times n-1$ and $n-1 \times m+1$ respectively, and the selection can take place after computing the updated Δ vector.

Assuming that the covariance function $\mathrm{kern}(\cdot,\cdot)$ of the GPRs remains the same after fitting the updated high-fidelity snapshots, the covariance matrix blocks $\mathbf{K}_{\mathrm{HH}}$ and $\mathbf{K}_{\mathrm{HL}}$ that appear in Equation (7) will contain larger entries in the rows and columns corresponding to the elements from the infill set. Thus, when queried on $\mathbf{x}$ from the surrogate PS, a more effective correction of the prior will be applied in Equation (9) and likewise a larger reduction of the variance will come from Equation (10). In practice, this infill criterion proves effective in generating a sparse sampling of the surrogate PS, and the rate of change of Δ_{i^*} provides a convergence criterion for early stopping of the multi-objective optimization.

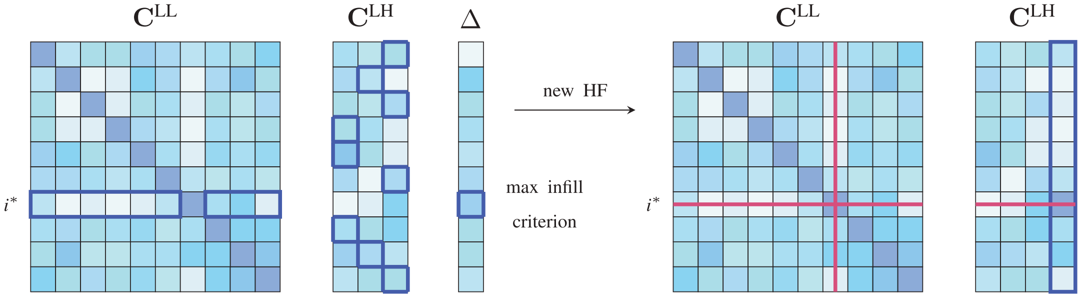

FIGURE 3 | On the left, position i^* in the low-fidelity PF is selected as maximizer of the infill criterion Δ evaluated on matrices $\mathbf{C}^{\mathrm{LL}}$ and $\mathbf{C}^{\mathrm{LH}}$ using Equation (19). The addition of the i^*-th sample to the high-fidelity set is simulated with the removal of row (and column) i^* from $\mathbf{C}^{\mathrm{LL}}$, its addition to $\mathbf{C}^{\mathrm{LH}}$, followed by the removal of row i^*. The next sample is selected using the updated matrices.

3.2.2 | Bayesian Optimization

BO aims to optimize an expensive black-box objective function, using a low number of function evaluations. The procedure builds a surrogate and updates it iteratively, selecting the next sample as the most promising for improving on the best-known configuration. A GPR is built on an initial set of objective function evaluations, and the selection of the samples is carried out by optimizing an easy-to-compute acquisition function, which leverages both the predicted value and the associated uncertainty [16]. The sampling of the parametric domain refines the GPR, thus effectively finding good configurations. This sampling balances between the so-called exploration (regions with high uncertainty) and exploitation (regions close to the current optimum).

The acquisition function represents the uncertain gain in terms of the objective function, so its maximization will select the next sample for the validation. Several formulations have been proposed, which vary in terms of exploratory or exploitative proclivity. Here, we briefly describe the negative lower confidence bound, the expected improvement, and the probability of improvement.

Negative lower confidence bound (NLCB) [44] is an exploratory acquisition function for minimization problems, expressed as

$$\alpha_{\mathrm{NLCB}}(\mathbf{x}) = -(\mu(\mathbf{x}) - \beta\sigma(\mathbf{x})), \tag{21}$$

where μ is the mean and σ is the standard deviation from the GPR posterior, with $\beta \geq 0$ weighting optimistically the contribution from the estimated uncertainty.

Expected improvement (EI) [45] is an exploratory acquisition function defined as

$$\alpha_{\mathrm{EI}}(\mathbf{x}, y^*) = \mathbb{E}[(y^* - \mu(\mathbf{x}))_+] = (y^* - \mu(\mathbf{x}))\Phi\left(\frac{y^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{y^* - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right), \tag{22}$$

where y^* denotes the best-known value of the objective function, ϕ and Φ are the probability density function and the cumulative distribution function, respectively, for the standard normal.

Probability of improvement (PI) [46] is an exploitative acquisition function defined as

$$\alpha_{\mathrm{PI}}(\mathbf{x}, y^*) = \Phi\left(\frac{y^* - \epsilon - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right), \tag{23}$$

where $\epsilon > 0$ prevents the acquisition function from selecting candidates excessively close to the best-known solution. A representation of a search step in BO, using different acquisition functions, is given in Figure 4.

In this paper, the objective function to minimize is the physical mass of the ship. We consider three contributions: the mass of the elements controlled by a decision variable, the mass of the elements that are fixed, and the mass of the reinforcement bars applied to correct buckling phenomena. We add a penalization term to the physical mass to handle constraints on the QoIs that do not have a practical expression in terms of the parameters. In particular, the designers fix the values for the acceptable number of yielded and buckled elements, represented by y_{crit} and b_{crit} respectively, so that the manual interventions on the optimal hull are limited in scope. However, the designers are open to consider configurations that slightly violate these constraints, if the gain in terms of physical mass is substantial. For this reason, we use quadratic penalization terms with coefficients tuned to balance the gain in terms of mass and the increase in yielding and buckling phenomena. The penalty takes smaller values close to the threshold, but renders the configuration sub-optimal as the penalized QoI grows.

The penalized objective function is obtained by summation of Equation (16) and a penalty term:

$$f(\mathbf{x}) = f_{\mathrm{mass}}(\mathbf{x}) + f_{\mathrm{pen}}(\mathbf{x}). \tag{24}$$

The penalization term $f_{\mathrm{pen}}(\mathbf{x})$ is defined as

$$f_{\mathrm{pen}}(\mathbf{x}) = c_{\mathrm{y}}(n_{\mathrm{y}}(\mathbf{x}) - y_{\mathrm{crit}})_+^2 + c_{\mathrm{b}}(n_{\mathrm{b}}(\mathbf{x}) - b_{\mathrm{crit}})_+^2, \tag{25}$$

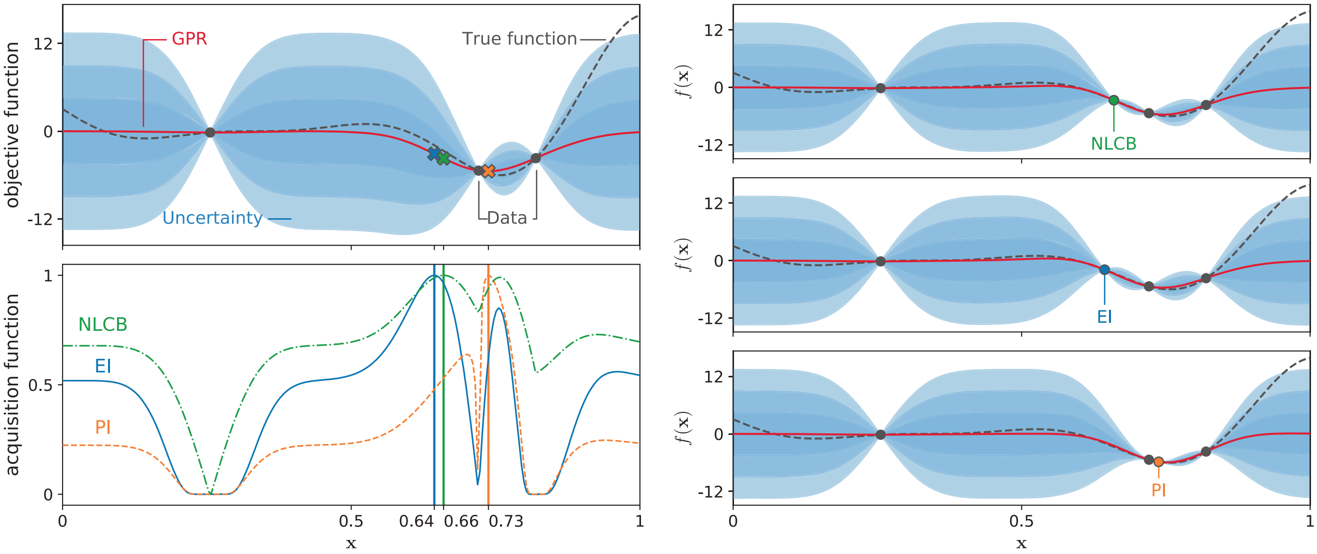

FIGURE 4 | On the left, the GPR prediction with its uncertainty on the top and the acquisition functions at the bottom. On the right, the updated GPR after the addition of the previously selected sample, for each acquisition function.

where c_{b} and c_{y} are some positive constants to be chosen by the designers, to quantify the trade-off between constraints violation and mass gain. These are conversion factors from the unit of measure of the penalized QoI, squared, to the unit of measure of the QoI being minimized. The quadratic penalization, although known to introduce non-stationarity in the BO [16], was preferred over more sophisticated techniques [16, 47, 48] since the designer can explicitly tune the trade-off between mass gain and violation. Equation (25) can be extended to include constraints on other QoIs, such as the vertical deflection.

We use the Emukit [49, 50] implementation of BO, where the optimization of the acquisition function is carried out by the `trust-constr` method from SciPy [51]. The `trust-constr` method leverages the algorithm from [52], which is a barrier method that uses sequential quadratic programming and trust regions to support generic inequality constraints. We leverage this capability to efficiently restrict the search space at runtime by producing a bound on the linear part of the penalized objective function.

Let $\mathbf{x}^*$ be the current best solution. Then, since $n_{\mathrm{b}}(\mathbf{x}) \geq 0$ and $f_{\mathrm{pen}}(\mathbf{x}) \geq 0$, a better optimum can only be found in the half-space where $\mathbf{x}$ satisfies

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) \geq m_{\mathrm{fixed}} + \mathbf{d} \cdot \mathbf{x}, \tag{26}$$

giving a linear inequality that will be updated each time $f(\mathbf{x}^*)$ decreases. This bound can be lax depending on the penalization term, but in practice, it is crucial to reduce the search space and achieve good optimization performances.

The constraint on the maximum VCG allows a linear inequality formulation as well, since Equation (17) is linear in $\mathbf{x}$. Then bounding from above with the critical value $\mathrm{VCG}_{\mathrm{crit}}$ gives

$$\frac{\mathrm{VCG}_{\mathrm{fixed}} m_{\mathrm{fixed}} + \sum_i \mathrm{VCG}_i \mathbf{d}_i \mathbf{x}_i}{m_{\mathrm{fixed}} + \sum_i \mathbf{d}_i \mathbf{x}_i} \leq \mathrm{VCG}_{\mathrm{crit}}, \tag{27}$$

$$\mathrm{VCG}_{\mathrm{fixed}} m_{\mathrm{fixed}} + \sum_i \mathrm{VCG}_i \mathbf{d}_i \mathbf{x}_i \leq \mathrm{VCG}_{\mathrm{crit}} \left(m_{\mathrm{fixed}} + \sum_i \mathbf{d}_i \mathbf{x}_i \right), \tag{28}$$

$$\sum_i (\mathrm{VCG}_i - \mathrm{VCG}_{\mathrm{crit}}) \mathbf{d}_i \mathbf{x}_i \leq (\mathrm{VCG}_{\mathrm{crit}} - \mathrm{VCG}_{\mathrm{fixed}}) m_{\mathrm{fixed}}. \tag{29}$$

Thus, our optimization problem is

$$
\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) = m_{\text{fixed}} + \mathbf{d} \cdot \mathbf{x} + m_{\text{bar}} n_{\text{b}}(\mathbf{x}) + c_{\text{y}}(n_{\text{y}}(\mathbf{x}) - y_{\text{crit}})_{+}^{2} + c_{\text{b}}(n_{\text{b}}(\mathbf{x}) - b_{\text{crit}})_{+}^{2}, \\
\text{subject to} \quad & m_{\text{fixed}} + \mathbf{d} \cdot \mathbf{x} \leq f(\mathbf{x}^{*}), \\
& \sum_{i} (\text{VCG}_{i} - \text{VCG}_{\text{crit}}) \mathbf{d}_{i} \mathbf{x}_{i} \leq (\text{VCG}_{\text{crit}} - \text{VCG}_{\text{fixed}}) m_{\text{fixed}}, \\
& \mathbf{x}_{i} \in \mathcal{D}_{i} \qquad \forall i \in I. \qquad (30)
\end{aligned}
$$

As an additional heuristic, we balance exploration and exploitation by switching between different acquisition functions after a number of iterations with no decrease in the current optimum.

The BO formulation used so far considers a continuous domain, but we are constrained to use commercially available thicknesses, resulting in a discrete domain. A naive implementation would optimize the acquisition and round to the closest domain point, but this approach is prone to repeated selection of the same configuration, and does not take into account the feasibility constraint on VCG. Multiple approaches can be found in the literature to handle ordinal and categorical variables with GPR, such as handling discretization inside the kernel [53, 54], while other approaches to BO replace the GPR with probabilistic models that handle discreteness transparently [55–57]. Other approaches perform optimization on a mixed variable domain, subject to both known and unknown constraints [58]. In our application, although the domain is fully discrete, the underlying physics is inherently continuous, and the QoIs exhibit discontinuities that are very small in relative value when the configuration is feasible, and no penalization is being applied. This observation allows us to rely on rounding, after gradient-based optimization of the acquisition under known constraints, instead of using derivative-free or sampling-based acquisitions. On detection of a duplicate sample, or if the sample is unfeasible, we solve an ILP [13] to find a new configuration that satisfies all constraints and has minimal distance from the original one.

Let $\bar{\mathbf{x}}$ be the current duplicate or infeasible candidate, I be the set of parameter indices, T be the set of possible parameter values, and x_{it} be the binary decision variable which takes value 1 when $\mathbf{x}_i = t$, and 0 otherwise. The next sample is retrieved by solving the following optimization problem

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in I} \sum_{t \in \mathcal{D}_i} x_{it} |t - \bar{\mathbf{x}}_i|^2, \\
\text{subject to} \quad & \sum_{t \in \mathcal{D}_i} x_{it} = 1, \qquad \forall i \in I, \\
& \sum_{i,t | \bar{\mathbf{x}}_i = t} x_{it} \leq |I| - 1, \\
& \sum_{i \in I} \sum_{t \in \mathcal{D}_i} x_{it} \mathbf{d}_i t \leq m_{\text{UB}} - m_{\text{fixed}}, \\
& \sum_{i \in I} \sum_{t \in \mathcal{D}_i} x_{it} (\text{VCG}_i - \text{VCG}_{\text{crit}}) \mathbf{d}_i t \leq (\text{VCG}_{\text{crit}} - \text{VCG}_{\text{fixed}}) m_{\text{fixed}}, \\
& x_{it} \in \{0, 1\}, \qquad \forall i \in I, \quad \forall t \in T, \qquad (31)
\end{aligned}
$$

where $\mathbf{d}_i$ is the linear density of the i-th parameter and m_{UB} the penalized mass of the current optimum. The first constraint ensures consistency, the second one guarantees that the new sample will not be a duplicate of $\bar{\mathbf{x}}$, and the rest replicate the constraint of the BO. If the newly found candidate is still a duplicate, we apply a random disturbance until an actual unvisited configuration is generated, or a maximum number of attempts is reached. For the solution of the ILP, we use the COIN-OR Branch-and-Cut solver (CBC) [59] through the Python-mip library [60]. This ILP has a small size, comprising only $|T||I|$ integer variables and $|I| + 3$ constraints, and is often solved to optimality by the pre-solving of the solver.

BO selects new candidate configurations until a computational budget is expended, given as the number of iterations or total execution time. The complete procedure is reported in Algorithm 3.

If the search finds no candidate that improves on the current best solution, the optimization is concluded. Otherwise, a high-fidelity simulation is required to validate such results, and the search is repeated. In practice, even with the available refinements to reduce the search space, BO on a large parameter space is only able to perform several rounds before failing to find new promising candidates.

ALGORITHM 3 | Specialized BO implementation.

```
Require: objective function $f$, high-fidelity parameters $\mathbf{X}$, and evaluations $\mathbf{y}$
Ensure: returns the configuration that minimizes $f$
  train GPR on $\mathbf{X}$, $\mathbf{y}$
  $\mathbf{x}^* \leftarrow \mathrm{argmin}_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})$
  $\alpha \leftarrow$ one of $\{\alpha_{\mathrm{NLCB}}, \alpha_{\mathrm{EI}}, \alpha_{\mathrm{PI}}\}$
  while number of iterations $\leq$ iter limit do
    $\mathbf{x} \leftarrow \mathrm{argmax}_{\mathbf{x}}\, \alpha(\mathbf{x})$ subject to Equation (26), Equation (29)
    $\mathbf{x} \leftarrow \mathrm{round}(\mathbf{x})$                ▷ could be provided by the BO framework
    while $\mathbf{x}$ is infeasible or $\mathbf{x} \in \mathbf{X}$ do
      $\mathbf{x} \leftarrow$ solution to ILP (31)
    end while
    if $f(\mathbf{x}) < f(\mathbf{x}^*)$ then
      $\mathbf{x}^* \leftarrow \mathbf{x}$
      update $f(\mathbf{x}^*)$ in Equation (26)
    end if
    $\mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{x}^*\}$
    $\mathbf{y} \leftarrow \mathbf{y} \cup \{f(\mathbf{x}^*)\}$
    train GPR on $\mathbf{X}$, $\mathbf{y}$
    if $f(\mathbf{x}^*)$ did not decrease in the last 100 iterations then
      $\alpha \leftarrow$ another one of $\{\alpha_{\mathrm{NLCB}}, \alpha_{\mathrm{EI}}, \alpha_{\mathrm{PI}}\}$
    end if
  end while
  return $\mathbf{x}^*$
```

3.2.3 | Principal Dimensions Search

To overcome the limitations of repeated BO on our large parameter space, we propose a more principled greedy heuristic, based on the exhaustive exploration of the neighborhood of the best candidate configuration. By leveraging the cheap surrogates, it is possible to evaluate all configurations that differ for one parameter from the chosen candidate. This approach is related to the cyclic coordinate search but performs a full scan of each parameter, thus the name PDS. The maximum number of surrogate evaluations is $\mathcal{O}(\sum_{i\in I} |\mathcal{D}_i| - 1)$, but in practice it is much lower due to the enforcement of Equation (26) and Equation (29). If no improvement is found, the search is concluded. Otherwise, the best configuration is used as the starting point of a new PDS until a computational budget is exhausted, and the best results are validated by the high-fidelity solver. The entire process is detailed in Algorithm 4.

This greedy heuristic is strongly exploitative of the current best solution and proves effective in finding new candidates, but is limited in its exploratory capabilities.

3.3 | Parameterization Refinement

The results of the optimization procedure are determined by the problem formulation, that is, by the parameterization chosen for the model. While a human designer is able to, in principle, assign the thickness of each element independently, the optimization of a model with so many decision variables would be impractical even for problems of moderate scale. On the other hand, a model that only uses a few parameters will be limited in its ability to balance the optimization of total mass and the reduction of structural failure phenomena, which are often quite local to critical geometrical features of the mesh. A first simplification comes from observing that, in the shipyard, the structure of the ship is assembled by metal sheets with dimensions about 2.5 m × 15 m, which correspond to a group of about 90 elements, on average. The elements that are part of the same sheet form a patch, and this is the finest level of parameterization that the designers can use in formulating the optimization problem. On the other hand, the coarsest parameterization consists of assigning a decision variable for each group of patches that shares the same regulatory minimum thickness. The common practice is to also take into consideration the orientation of the patches and the possible criticalities emerging from the stress configurations under the load conditions. Upon convergence of the optimization pipeline, the designer analyzes the optimum configuration to identify whether it is possible to improve upon its results. A common situation is that a large

ALGORITHM 4 | Principal direction search.

```
Require: objective function $f$, high-fidelity parameters $\mathbf{X}$
Ensure: returns a feasible solution through PDS
  $\mathbf{x}^* \leftarrow \mathrm{argmin}_{\mathbf{x}\in\mathbf{X}} f(\mathbf{x})$
  while number of iterations ≤ iter limit and elapsed time ≤ time limit do
     $\mathbf{x}^{\mathrm{base}} \leftarrow \mathbf{x}^*$
     $\mathbf{x} \leftarrow \mathbf{x}^{\mathrm{base}}$
     for i in I do
        for $t \in \mathcal{D}_i \setminus \{\mathbf{x}_i^{\mathrm{base}}\}$ do
           $\mathbf{x}_i \leftarrow t$
           if $\mathbf{x}$ is feasible and $f(\mathbf{x}) \leq f(\mathbf{x}^*)$ then
              $\mathbf{x}^* \leftarrow \mathbf{x}$
           end if
        end for
        $\mathbf{x}_i \leftarrow \mathbf{x}_i^{\mathrm{base}}$
     end for
  end while
  return $\mathbf{x}^*$, $f(\mathbf{x}^*)$
```

group of patches, controlled by the same decision variable, is assigned a high thickness even if only a small, localized subgroup of those patches would incur yielding and buckling. The designers would then split the original grouping of the patches to manually assign different thickness values to the subgroups. A lower thickness would be assigned to the subgroup that does not incur structural issues, and a higher thickness to the one where the failure phenomena are localized. This procedure could result in lowering the total mass of the model, depending on the difference in thickness and the extent of the structural criticalities. It would be natural to derive a new optimization problem where the two subgroups of patches are assigned to two different decision variables, but this approach would incur two expensive issues: the setup of a new problem is a cumbersome operation, and the new optimization would require a new initial sampling and surrogate construction.

We propose an automatic refinement of the parameterization based on the solution of a set of ILPs, which will determine a division of each group of elements according to its structural response. This approach has been inspired by a similar idea in the context of multiscale finite elements [61], where the number of finer, expensive problems to be solved was reduced by clustering the boundary conditions applied to the sub-elements.

In our case, we start from the best solution obtained during the optimization and the set of surrogates constructed based on the high-fidelity evaluations collected during the pipeline execution. A number of configurations are evaluated in the vicinity of the best solution to obtain the yielding and buckling states of each element. For each parameterized section, an ILP is constructed to choose the best parameter value for each patch, so that an objective function similar to the one used in BO is minimized. The resulting thickness assignments identify the optimal refined parameterization.

Let P_i be the set of patch indices assigned to the i-th parameter $\mathbf{x}_i$, and $\mathcal{D}_i$ the set of possible parameter values. Additionally, let y_{pt} be the number of elements in patch $p \in P_i$, when $\mathbf{x}_i = t$, that yield under any of the load conditions, and let b_{pt} be the analogue quantity for buckling. We assume that each patch is independent of the rest of the ship, so that y_{pt} and b_{pt} combine linearly and can be computed (using surrogates and high-fidelity evaluations when available) by enumerating $t \in T$ and keeping the rest of the parameter values fixed. These evaluations follow the same order as in Algorithm 4.

The binary decision variables x_{pt} express the assignments of patches to parameter values, with $x_{pt} = 1$ only if patch p has thickness t. For consistency, for each patch p, it is required that exactly one of the x_{pt} variables is nonzero. The upper bound on the VCG takes the same form used in Equation (29). For the objective function, we aim to minimize the total mass of the patches in P_i, their reinforcement bars, and penalty terms given by the sum of squares of y_{pt} and b_{pt}. Finally, we fix the actual number of different parameter values $n_{\mathrm{clusters}} \geq 2$ through the additional decision variables u_t and their consistency constraints.

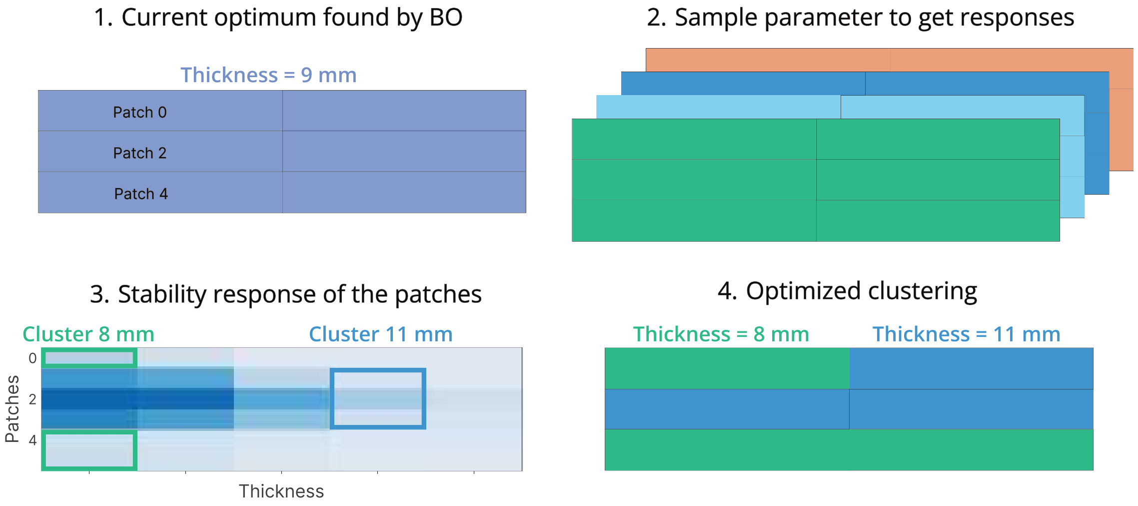

FIGURE 5 | The reparameterization procedure on a simplified parameterized section. The section is composed of six patches, represented in steps 1, 2, and 4 with different colors corresponding to different thicknesses. In step 1, all patches have the same thickness as selected by the previous optimization. In step 2, different thicknesses are evaluated to obtain the structural responses of each patch. In step 3, the structural responses determine the reparameterization problem, and its optimal solution clusters the patches in two groups. In step 4, the patches are assigned the optimized thicknesses, and two parameterized sections are determined.

We obtain the following optimization problem:

$$
\begin{aligned}
&\text{minimize} && \sum_{p\in P_i}\sum_{t\in \mathcal{D}_i} x_{pt}(\mathbf{d}_p t + m_{\text{bar}} b_{pt} + c_{\text{y}} y_{pt}^2 + c_{\text{b}} b_{pt}^2), && \\
&\text{subject to} && \sum_{t\in \mathcal{D}_i} x_{pt} = 1, && \forall p\in P_i, \\
& && x_{pt} \le u_t, && \forall p\in P_i,\quad \forall t\in \mathcal{D}_i, \\
& && \sum_{p\in P_i} x_{pt} \ge u_t, && \forall t\in \mathcal{D}_i, \\
& && \sum_{t\in \mathcal{D}_i} u_t = n_{\text{clusters}}, && \\
& && \sum_{p\in P_i}\sum_{t\in \mathcal{D}_i} x_{pt}(\text{VCG}_p - \text{VCG}_{\text{crit}})\mathbf{d}_p t \le (\text{VCG}_{\text{crit}} - \text{VCG}_{\text{res}}) m_{\text{res}}, && \\
& && x_{pt} \in \{0,1\}, && \forall p\in P_i,\quad \forall t\in \mathcal{D}_i, \\
& && u_t \in \{0,1\}, && \forall t\in \mathcal{D}_i,
\end{aligned}
\tag{32}
$$

where $\mathbf{d}_p$ is the linear density of patch p, VCG_p is the VCG of patch p, and VCG_{res} and m_{res} are the VCG and mass, respectively, of all elements not controlled by the current parameter.

The total number of integer variables is $(|P_i| + 1)|\mathcal{D}_i|$ and the number of constraints is $|P_i||\mathcal{D}_i| + |P_i| + |\mathcal{D}_i| + 2$. This class of problems is NP-hard, but branch and cut simplex-based solvers are extremely efficient in practice thanks to pre-solve transformations and branching rules [62, 63]. An example of the procedure, for a simplified section composed of six patches, is depicted in Figure 5.

We remark that the objective functions in ILP (32) and Equation (24) are fundamentally different in the way they handle the penalization of yielded and buckled elements, as the former uses a sum of squares and the latter uses the square of a sum.

In ILP (32), the objective function can be split into the contributions from each patch to the physical mass, due to thickness and additional stiffeners, and the penalization of failed elements as

$$\sum_{p\in P_i}\sum_{t\in D_i} x_{pt}(\underbrace{\mathbf{d}_p t}_{\text{patches mass}} + \underbrace{m_{\text{bar}} b_{pt}}_{\text{reinforcement bars}} + \underbrace{c_y y_{pt}^2 + c_b b_{pt}^2}_{\text{per-patch penalty}}), \tag{33}$$

which is linear in the decision variables x_{pt}. The proper reformulation of Equation (24) in terms of x_{pt} requires instead the quadratic programming expression

$$\sum_{p\in P_i}\sum_{t\in D_i} x_{pt}(\underbrace{\mathbf{d}_p t}_{\text{patches mass}} + \underbrace{m_{\text{bar}} b_{pt}}_{\text{reinforcement bars}}) + f_{\text{pen}}(x), \tag{34}$$

where the penalty term for the binary decision variables, corresponding to Equation (25), is

$$f_{\text{pen}}(x) = \underbrace{\left(\left(\sum_{p\in P_i}\sum_{t\in D_i} x_{pt} c_y y_{pt}\right) - y_{\text{crit}}\right)_+^2}_{\text{global yielding penalty}} + \underbrace{\left(\left(\sum_{p\in P_i}\sum_{t\in D_i} x_{pt} c_b b_{pt}\right) - b_{\text{crit}}\right)_+^2}_{\text{global buckling penalty}}. \tag{35}$$

The thresholds y_{crit} and b_{crit} in Equation (35) need to be adapted to the current i-th parameterized section, as Equation (25) uses global values. One option would be subtracting the yielded and buckled elements' totals of the other sections from the critical values, but the combination of the separate problems could still exceed the global constraints. As an alternative, ILP (32) needs to be reformulated as a global optimization problem, optimizing all the parameterized sections at once, which is impractical for large models. In practice, the usage of ILP (32) leads to a more severe penalization of failure phenomena in each patch, so that the updated model will offer better trade-offs during the optimization.

By solving ILP (32) for each parameterized section, and possibly multiple n_{clusters}, a collection of candidate reparameterizations is obtained. At most one refinement can be applied to each parameterized section, including the trivial one, which leaves it unchanged. Moreover, the designers could specify a maximum number of parameters for the model. The selection of the optimal set of refinements can be obtained by the solution of a knapsack ILP [13], where the cost of a refinement is the number of clusters and its value is given by ILP (32).

Since the construction of the VCG constraint assumes that all the elements outside the current group of patches do not change mass, the parameter configuration obtained by joining the optimized assignments might violate the global VCG constraint. Indeed, the VCG constraint in each local ILP (32) has slack dependent on the current optimum, so in an extreme case, each section with VCG above the global threshold could increase its VCG contribution, and the opposite for sections with VCG below the global threshold. The actual changes in VCG contribution could then violate the global VCG constraint. Even so, the goal of the procedure is to obtain an expressive parameterization: the locally-optimal thickness assignments will be overwritten by the following optimization procedure, which enforces the global VCG constraint.

Once the optimal selection of clusters has been chosen, the i-th parameterized section associated to $\mathbf{x}_i$ is also partitioned in a set of patch clusters $\{\mathcal{P}_j\}_{j=1}^m$ with $1 \le m \le n_{\text{clusters}}$. The patches in each cluster $\mathcal{P}_{j>1}$ will be assigned to a new parameter $\mathbf{x}_k$, with feasible domain $\mathcal{D}_i$, decreasing the number of patches assigned to $\mathbf{x}_i$. With the new parameterization, the configurations generated with the coarse model can still be retrieved by enforcing the constraints

$$\mathbf{x}_i = \mathbf{x}_k \qquad \forall k \in \mathcal{K}_i, \tag{36}$$

where $\mathcal{K}_i$ collects the indices of the $m-1$ parameters generated for the clusters $\mathcal{P}_{j>1}$. This feature is crucial for the efficiency of the optimization procedure, as all the previous high-fidelity simulations need not be executed again. On the other hand, all the high-fidelity samples collected so far can be viewed as coming from a constrained version of the current parameter space, where Equation (36) reduced the feasible configurations. Due to this consideration, the surrogates must be rebuilt on a sampling that includes a configuration where Equation (36) is not active, so that the effects of the new parameterization are properly observed. For this task, we repeatedly perform a random sampling of the new parameter space, keeping the candidate set that maximizes the cumulative distance between the high-fidelity samples and the new ones.

At this point, the surrogates are rebuilt on the updated high-fidelity samples, and the optimization, both multi- and single-objective, can take place on a more expressive model. With this approach, the parameterization is adapted as the optimization progresses and is thus able to overcome the biases that the designers could have introduced in the initial model creation. This presents a critical advantage when the hull features high complexity due to novel structures or design constraints, and enables the pipeline to provide high-quality initial designs with minimal supervision by the users.

4 | Numerical Results

This section presents the results from the application of our optimization pipeline to two ship models: a simplified midship section typical of the initial design phase in Section 4.1, and a full ship model with all environments and functional features in Section 4.2.

4.1 | Midship Section

For the initial development and testing of our methods, we chose a simplified model of the main section of a typical hull. The model is depicted in Figure 6. It consists of 75,192 elements, of which 52,360 are assigned to a parameterized section as reported in Table 1. The parameterized elements are further grouped in 582 patches, by first defining a 3D grid with cell dimensions that follow a typical patch disposition, and then grouping the elements from the same structure (such as the same deck) whose centroids fall in the same cell. The initial parameterization follows the orientation of the patches and the regulatory minimum thickness of the structural members. Thus, lower bounds (LBs) for the parameter values come from standard regulations, while the upper bounds have been chosen according to common design practices. The total number of configurations in the domain is 62,720. This model is a quarter of the actual hull: during the high-fidelity simulations, it is reflected across the xz and yz planes.

A NASTRAN run on this model takes about 1 min, and the query time of the surrogates is 0.15 s, for a speedup factor of 400. The maximum value for the VCG is 15 m. In this experiment, the critical values for the yielding due to the stresses and the von Mises criterion are lower than the regulatory standard, as the simplicity of this model would otherwise result in no yielded elements. For this model, we choose 20 as the maximum number of parameterized sections that can be defined.

We start the optimization pipeline by performing a random sampling of the parameter space, selecting 20 configurations in addition to the default one. We adopt a maximin design [16]: for a set number of iterations, we draw a set of configurations and measure the minimum of their Euclidean distances. The set with the highest minimum distance is retained and evaluated with the high-fidelity solver. We remark that the choice of a small initial sampling is motivated by the

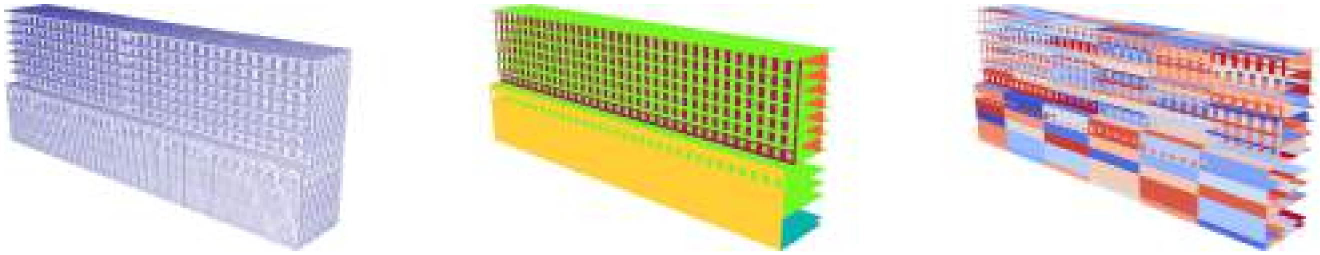

FIGURE 6 | Full view of the midship section on the left, the initial parameterization in the middle, and the details of the parameterized patches on the right.

TABLE 1 | Parameters description of the midship section test case. Thickness values are in mm.

Parameter	Region	Patches	Default	Min.	Max.	# values
$\mathbf{x}_1$	Bottom and inner bottom	72	14.0	12.0	20.0	14
$\mathbf{x}_2$	Decks from 2 to 12	396	5.0	5.0	15.0	5
$\mathbf{x}_3$	External bulkheads	30	10.0	8.0	15.0	14
$\mathbf{x}_4$	Internal bulkheads	42	5.0	5.0	15.0	8
$\mathbf{x}_5$	Shell plating	42	8.0	8.0	15.0	8

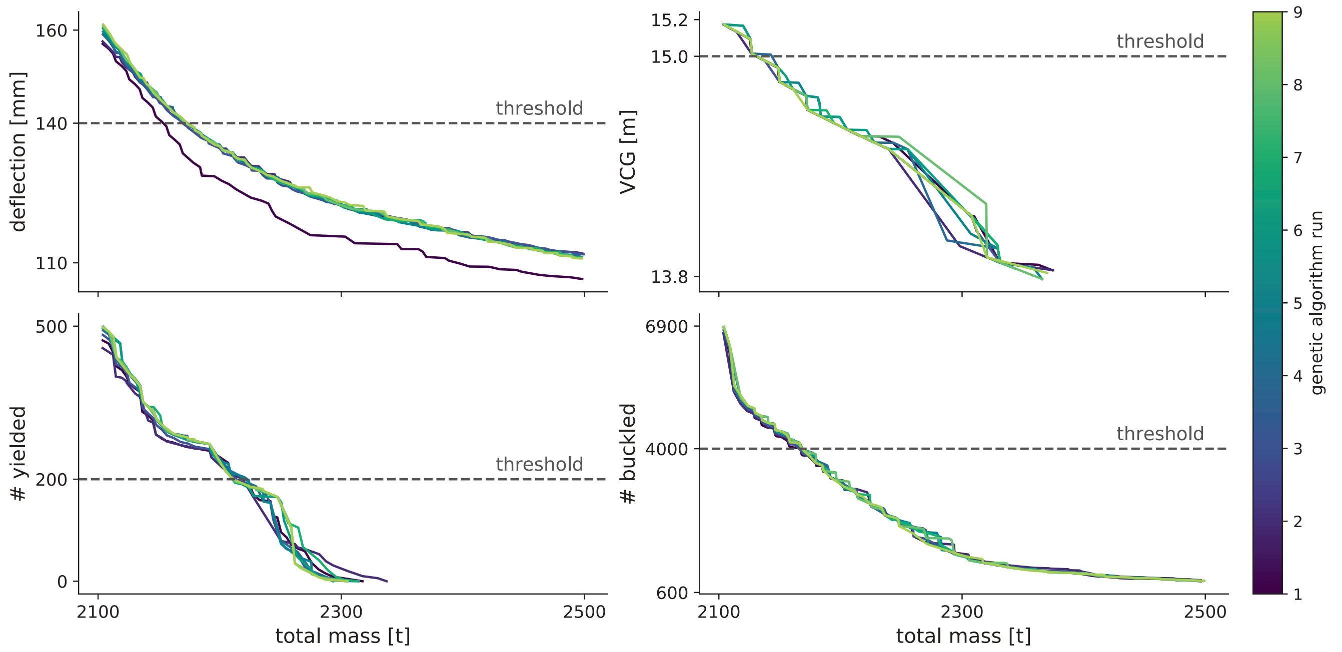

FIGURE 7 | Evolution of the surrogate PF in subsequent runs (generations) of the genetic algorithm, for the midship section with 5 parameters. Each figure depicts the 2-dimensional frontier computed from the projection of the full 5-dimensional frontier.

subsequent multi-objective optimization, in which high-fidelity experiments are performed on parameter configurations sampled by the surrogate PF. The truncation rank chosen for the POD is 6, so that the discarded modes have a normalized module smaller than 0.01. This is a relatively higher ratio than what is usual in the POD literature, but finds its justification in that we are not interested in the stress field per se, but in sufficiently accurate modeling of the number of yielded and buckled elements. In Appendix C, the truncation rank and the resulting errors on the QoIs are discussed in detail. The GPRs for the reduced coefficients use the squared exponential kernel with ARD and epistemic noise variance for numerical stability.

The multi-objective optimization enables the designers to inspect the effective trade-offs between different QoIs, so that the critical thresholds for the constraints can be determined from a global perspective. Figure 7 shows the progress of the surrogate PF as the pipeline repeats a sequence of GA run, infill selection and high-fidelity validation, and surrogates update. The population size is set to 2000 configurations, the number of generations is 10, and the number of samples selected by the infill criterion is 9. For the midship section, the acceptable numbers of yielded and buckled elements, and the maximum vertical deflection, are set to 200, 4000, and 140 mm, respectively.

Single-objective mass optimization starts with BO and a time limit of 5 min. The history of the best surrogate candidates is presented in Figure 8 in terms of the percentage gap from the theoretical LB, that is,

$$m_{\text{gap}}(\mathbf{x}) = 100\frac{\mathbf{d}\cdot(\mathbf{x}-\mathbf{x}_{\text{LB}}) + m_{\text{bar}}n_{\text{b}}(\mathbf{x}) + f_{\text{pen}}(\mathbf{x})}{\mathbf{d}\cdot\mathbf{x}_{\text{LB}}}. \tag{37}$$

This quantity highlights the effect on the mass actually controlled by the parameters and does not depend on how extensively the model has been parameterized. In this case, $m_{\text{fixed}} = 1108.21$ t and $\mathbf{d}\cdot\mathbf{x}_{\text{LB}} = 975.55$ t.

Only the first round is able to find better candidates, which are then confirmed by the following high-fidelity experiment. The second BO execution does not improve on the previous, nor do the subsequent PDS refinements.

At this point, the optimum configuration is analyzed for the parameterization refinement, as in Section 3.3. For each parameter, the number of possible clusters is set to 2. In Figures 9 and 10, we focus on the internal bulkheads controlled by $\mathbf{x}_4$. The optimal clustering selects thicknesses 6 and 9 mm, which, when applied to the entire parameter, give the buckling patterns in Figure 9. Finally, Figure 10 shows the assignment of elements to the two clusters. Notably, the three patches in the top right corner were assigned a lower thickness value: the ILP solution leverages the trade-off in using

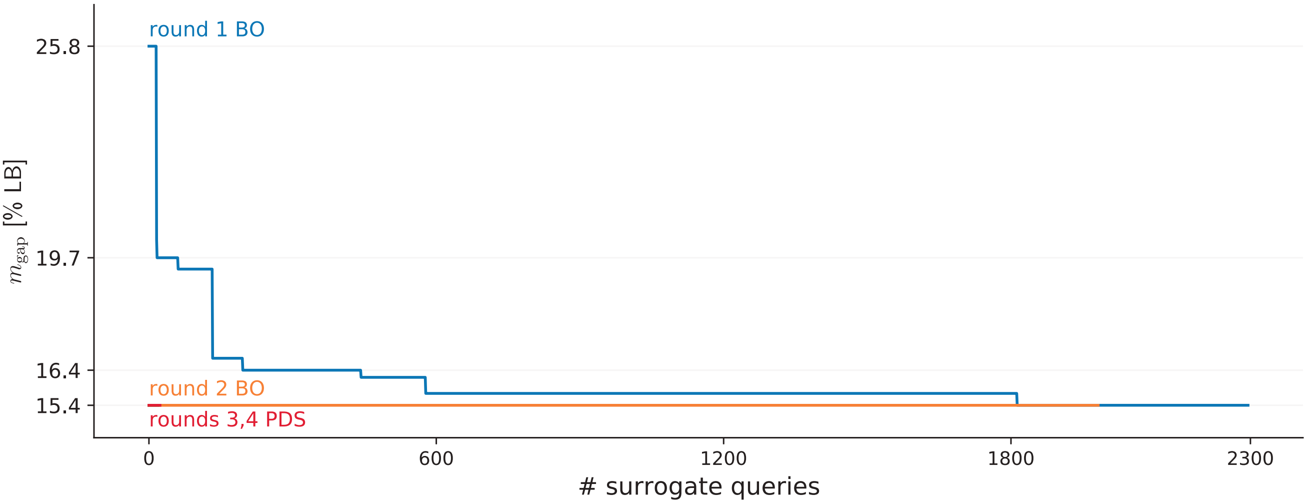

FIGURE 8 | History of the surrogate best solution during the mass optimization of the midship section, 5 parameters. After the end of each round, the best configurations found are validated by a high-fidelity simulation, and the surrogates are rebuilt on the updated database.

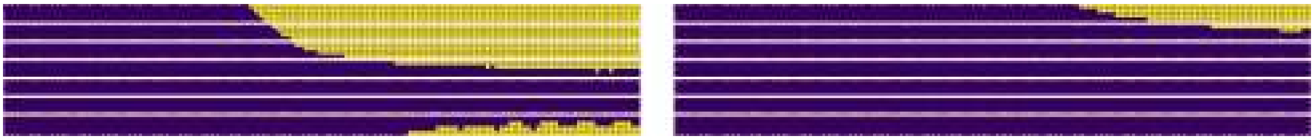

FIGURE 9 | Buckling state of the elements controlled by $\mathbf{x}_4$, with failed elements in yellow and healthy elements in purple. The state on the left is obtained for $\mathbf{x}_4 = 6$ mm, the one on the right for $\mathbf{x}_4 = 9$ mm.

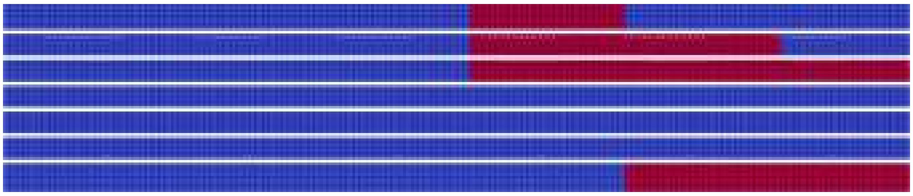

FIGURE 10 | Optimal clustering of the elements controlled by $\mathbf{x}_4$. The elements in red will be associated to a new decision variable, while the elements in blue remain associated to $\mathbf{x}_4$.

reinforcement bars instead of thicker, heavier patches. Figure B1 shows the reparameterization of all the sections in the model.

After the update of the parameterization, the situation is analogous to an initial problem formulation, but with many high-fidelity configurations already available. However, these configurations were generated from a coarser parameterization, and are not representative of the interactions between the newly created parameters. To create the surrogate models on the new parametric domain, we generate 20 configurations by random sampling from a uniform distribution on the updated domain. We produce multiple candidate sets and select the one maximizing the minimum distance between the candidates, and between candidates and high-fidelity samples. The candidate set is passed to the high-fidelity solver to generate the corresponding snapshots, and the surrogates for the updated parametric domain are created. At this point, the pipeline resumes with multi-objective and single-objective optimization as done previously, but with a higher number of decision variables and a more expressive problem formulation. The sequence of alternating optimization and parameterization refinement can be repeated until a set number of parameters is reached, or the prospective decrease in objective function results in diminishing returns.

For the midship section, we perform two reparameterization steps, the first creating 5 new parameters for a total of 10, and the second creating 7 new parameters for a total of 17. Each time, the truncation rank of the POD is increased by

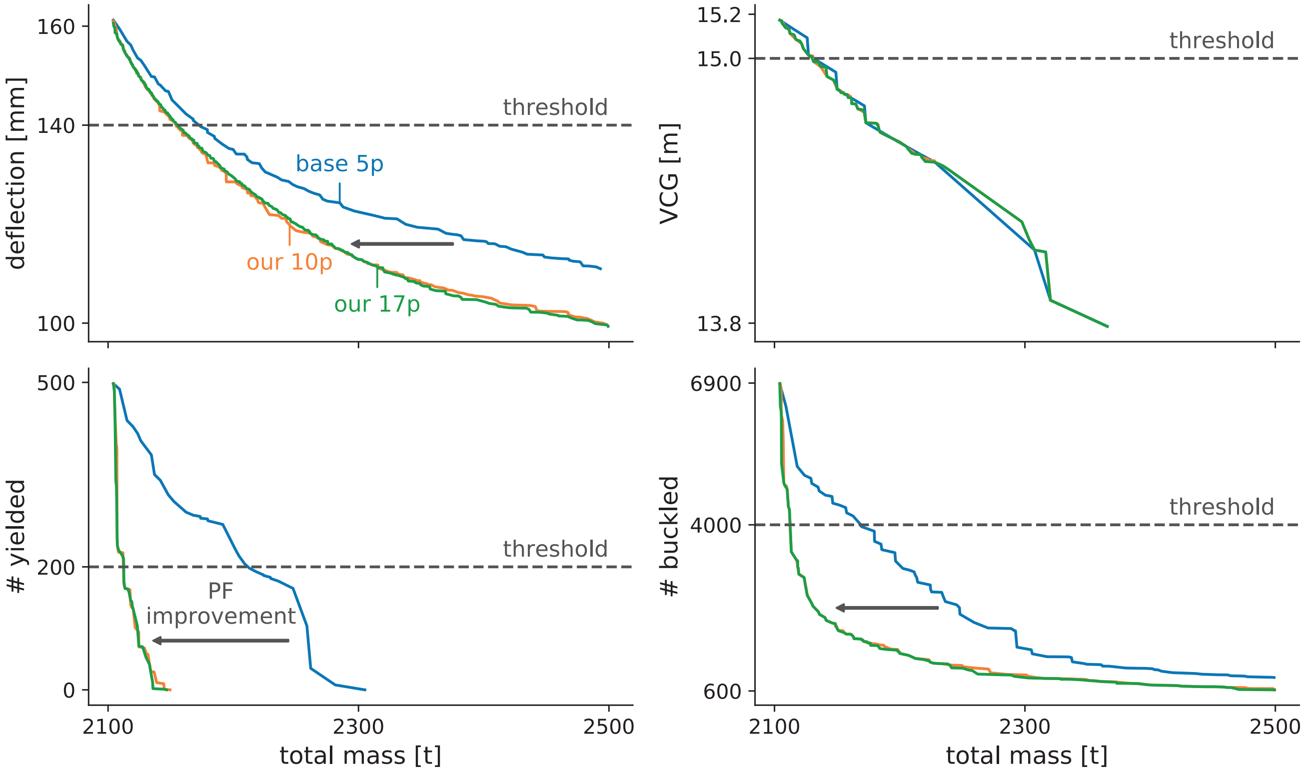

FIGURE 11 | Evolution of the surrogate Pareto frontier for the midship section, for different parameterizations.

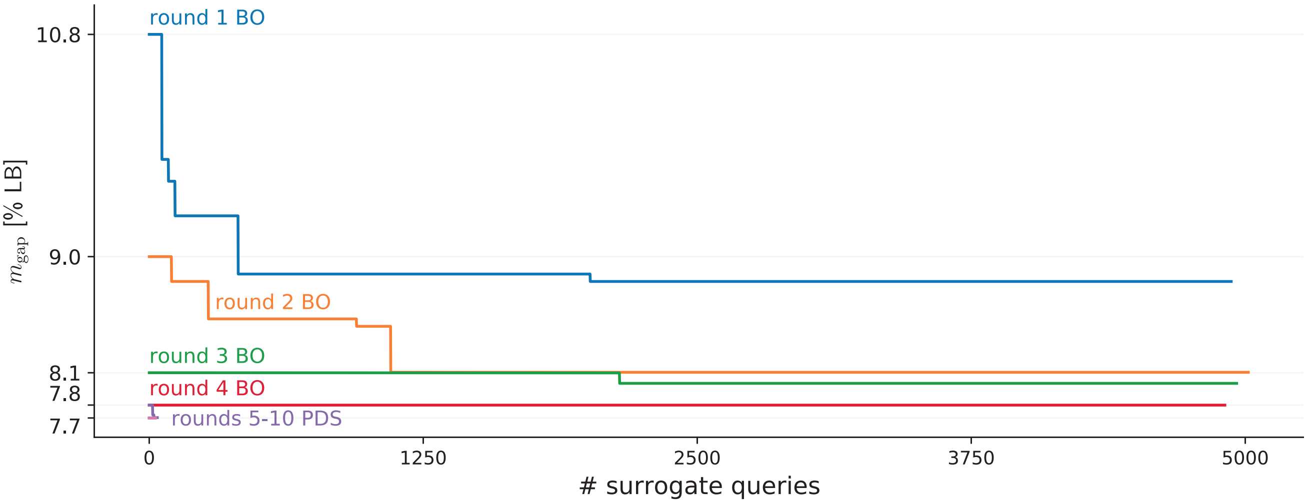

FIGURE 12 | History of the surrogate best solution during mass optimization of the midship section with 10 parameters. After the end of each round, the best configurations found are validated by a high-fidelity simulation, and the surrogates are rebuilt on the updated database.

the number of added parameters. Figure 11 shows the differences in the final PFs as the number of parameters increases. The largest improvement in PF quality is attributed to the first reparameterization, as seen in the plots for the number of yielded and buckled elements. The updated PF intercepts the thresholds at a much lower total mass value, indicating that the subsequent single-objective optimizations could find much lighter configurations with no additional drawbacks. The second refinement does not produce an appreciable change in the PFs, and thus the procedure is stopped. Figure 12 shows the single-objective optimization of the 10-parameter problem. The finer parameterization enables the optimizer to find better configurations, with the PDS being repeated several times. In this case, the high-fidelity validation disproves a large number of optimum candidates.

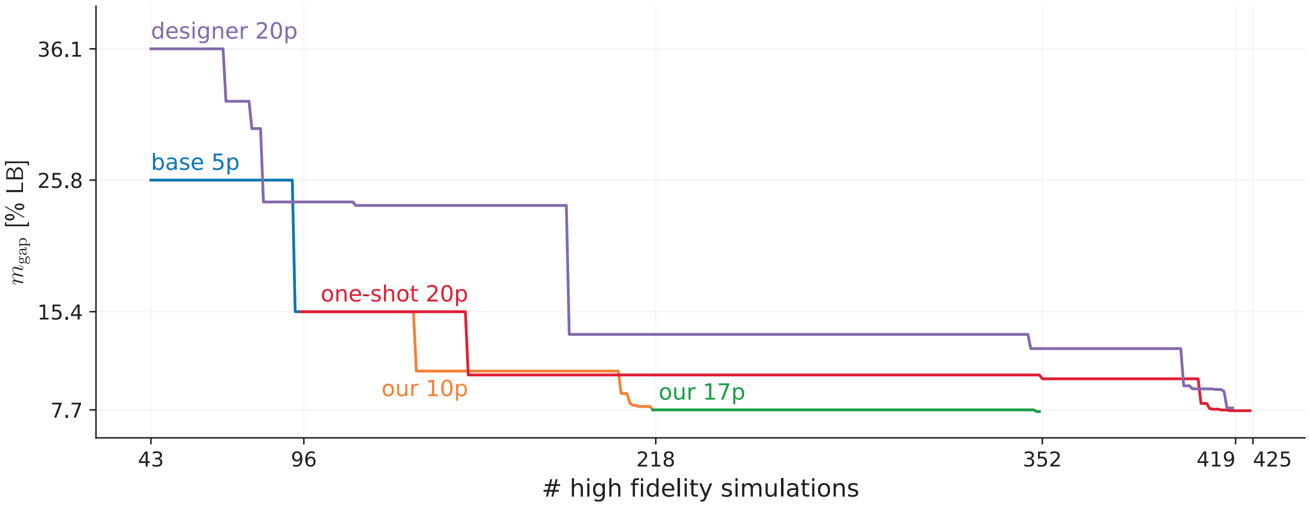

FIGURE 13 | Evolution of the optimal high-fidelity configuration for different parameterizations of the midship section.

TABLE 2 | Optimization results for different parameterizations of the midship section. The thresholds for the number of yielded and buckled elements are 200 and 4000, respectively. The threshold for the deflection is 140 mm and the maximum VCG is 15 m.

Parameterization	m_{gap} [% LB]	#yielded	#buckled	Deflection [mm]	VCG [m]
Initial configuration	> 1000	395	6211	150	14.79
Base 5 p.	15.44	196	3994	138	14.93
Our method 10 p.	7.70	33	3984	140	14.99
Our method 17 p.	7.57	11	3563	140	14.99
Our one-shot 20 p.	7.64	5	2729	140	14.99
Designer 20 p.	7.86	22	4004	140	14.99

We also test the effect of a single reparameterization exhausting the parameters budget, which we refer to as “one-shot,” by generating 15 new parameters from the initial 5. Finally, we optimize a 20-parameter model produced by the designers to compare the efficacy of our method. The POD truncation ranks for both cases are 21.

The evolution of optimal high-fidelity configurations for all the approaches is shown in Figure 13. The large sequences of nondecreasing ships are due to the multi-objective optimization, in which the penalized mass is only incidentally optimized, and the focus is on obtaining better exploration of the PF. Table 2 collects the results of the optimum for each problem formulation. Overall, the largest parameterizations achieve similar results in terms of gap from the LB, but with a number of differences. Iterative reparameterization shows a much faster decrease, with the 10-parameter model reaching a comparable objective value to the others, but with a much lower number of high-fidelity simulations. The second reparameterization, with 17 parameters, decreases the optimum only by a negligible amount. However, we observe that the number of buckled elements is about 11% lower than the threshold value, suggesting that further rework by a human designer could benefit from this buffer from the constraint’s critical value. The one-shot parameterization requires a larger number of high-fidelity simulations in the multi-objective phase, but the buffer in terms of buckled elements is even more pronounced than for the 17-parameter model, being about 32% of the threshold value. The designer-provided reference requires a number of high-fidelity evaluations similar to the one-shot case, but the final configuration reaches the threshold on the number of buckled elements.

4.2 | Full Ship

The full-scale test model in Figure 14 comes from a ship developed by Fincantieri SpA. This model consists of 485,736 elements, of which 270,656 are linked to the parameters in Table 3 and organized in 5005 patches. The MSC NASTRAN solution time for this model is 7 min, and the surrogate prediction time is 0.5 s, for a speedup factor of 840. The default

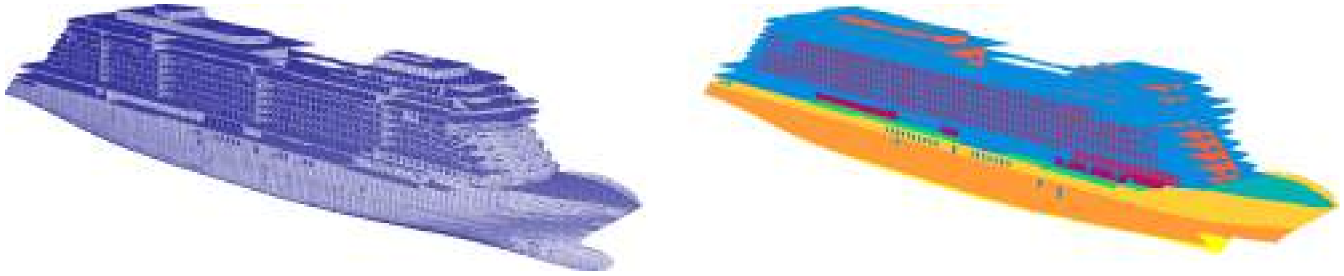

FIGURE 14 | Full view of the ship on the left, initial parameterization on the right.

TABLE 3 | Parameters description of the full ship test case. Thickness values are in mm.

Parameter	Region	Patches	Min.	Max.	# values
$\mathbf{x}_1$	Decks in public areas	3038	5.0	25.0	16
$\mathbf{x}_2$	Decks in machinery areas	548	6.5	25.0	13
$\mathbf{x}_3$	Lifeboat exposed deck	90	13.0	25.0	5
$\mathbf{x}_4$	Inner bottom	252	12.0	25.0	6
$\mathbf{x}_5$	Bottom	312	15.0	25.0	3
$\mathbf{x}_6$	Shell plating, above waterline	60	14.0	25.0	4
$\mathbf{x}_7$	Shell plating, below waterline	118	15.0	25.0	3
$\mathbf{x}_8$	Internal bulkheads	367	5.0	25.0	16
$\mathbf{x}_9$	External bulkheads, in super-structures	190	8.0	25.0	10
$\mathbf{x}_{10}$	External bulkheads	31	6.5	25.0	13

parameter configuration corresponds to the minimum regulatory thickness of the parameterized sections, while all the upper bounds have been left to the maximum commercially available value. The parameters are equally divided into two groups, one containing decks and bulkheads having a high number of feasible values, and the rest of the structural elements with a small domain. The resulting domain contains 467 million unique combinations. Due to the increased complexity of this model, the maximum number of parameters is 40. To comply with the proprietary constraints set by the industrial partner, we present the QoIs in a relative format, and the critical values are not shown.

The initial sampling of the parameter space is limited to 20 configurations other than the default one. The truncation rank chosen for the POD is 16, and the GPRs use the same structure as for the midship section case. Details on the truncation ranks and the corresponding errors can be found in Appendix C. As before, multi-objective optimization is carried out in order to identify the appropriate thresholds for the number of yielded and buckled elements. The maximum value of VCG is determined by the model geometry, while no constraint on the vertical deflection is specified. We repeat the same tests from Section 4.1: a sequence of iterative reparameterizations from 10 to 40 parameters, a one-shot reparameterization from 10 to 40 parameters, and the optimization of a 40-parameter model provided by the designers.

In the iterative reparameterization test, the refinement step is performed 3 times, giving models with 16, 27, and 40 parameters each. Figure 15 shows the evolution of the surrogate PF. Each refined model improves the previous PF, with the largest effect being achieved on the number of buckled elements. The first reparameterization shows a more pronounced reduction of the objective, while the third appears to have a limited effect. The threshold values for the number of yielded and buckled elements result in much more stringent conditions than those used in Section 4.1.

The evolution of optimal high-fidelity configurations for all the approaches is shown in Figure 16, and the details of the optimum configuration for each problem is shown in Table 4. A striking difference from Figure 13 is the large decrease in objective function due to the reparameterizations to 16 and 40 parameters, showing that the ILP was able to generate a highly performant and feasible configuration. As in the case of the PFs, refinements beyond the first incur lower gains, and the final refinement of the incremental approach only decreases the optimum value by around 1% of the LB defined in Equation (37). Multi and single-objective optimization of the one-shot model are not able to achieve as large improvements as the initial ILP-optimal configuration, with the final result being worse than the 27-parameter model by about 1% of the

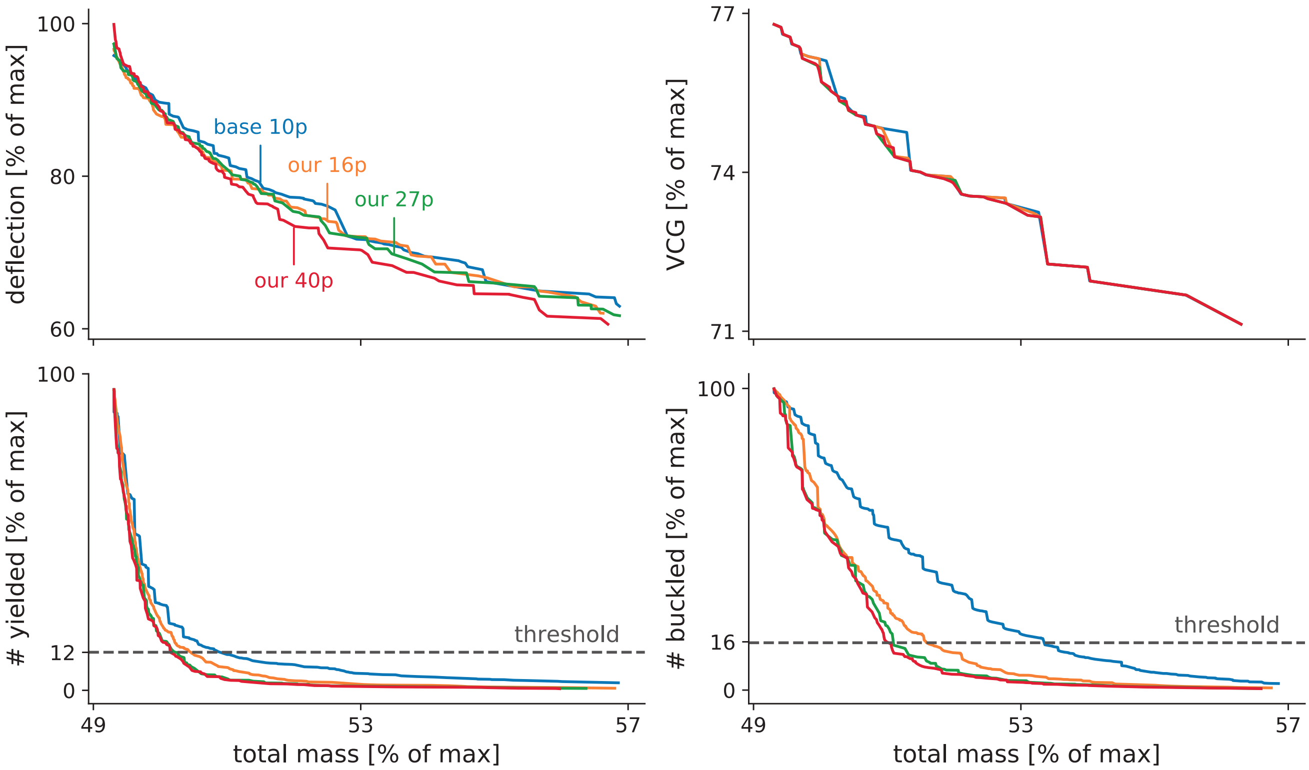

FIGURE 15 | Evolution of the surrogate Pareto frontier for the full ship, for different parameterizations. The threshold for the VCG constraint is not shown, since the PF of the projection lies well below it.

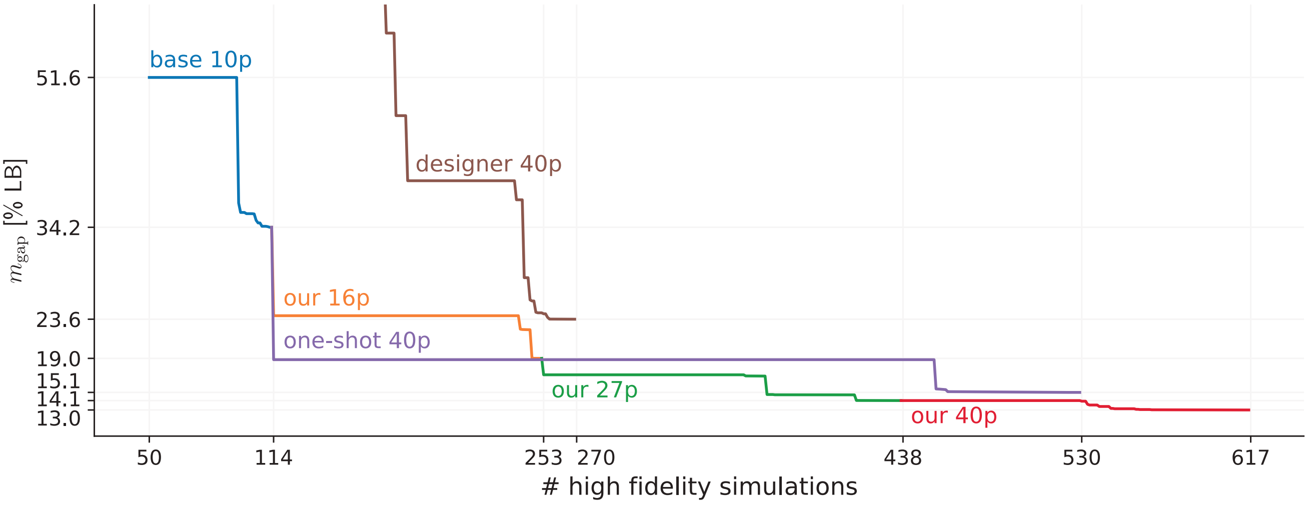

FIGURE 16 | Evolution of the optimal high-fidelity configuration for different parameterizations of the full ship.

LB. The final configuration of the designer model is better than the 10-parameter model and reaches its optimum with a lower number of high-fidelity simulations than with the other approaches, but it is only marginally better than the ILP optimum that generated the 16-parameter model. All the parameterizations generated through our automated procedure were able to outperform the one from the designers. These results suggest that the high complexity of the full ship, with multiple large parameterized groups of patches, provided a harder challenge in designing an effective parameterization, even for an expert.

Regarding the constrained QoIs, the number of buckled elements and VCG are close to the threshold for all instances. The number of yielded elements does so only for the models that were iteratively refined to 27 and 40 parameters, while in the other instances, it remains around half of the threshold value. This behavior is different from what was observed

TABLE 4 | Optimization results for different parameterizations of the full ship.

Parameterization	m_{gap} [% LB]	# yielded [% crit]	# buckled [% crit]	VCG [% crit]
Initial configuration	> 1000	789.75	636.38	96.26
Base 10 p.	34.21	46.25	100.38	99.35
Our method 16 p.	19.01	56.00	99.88	99.76
Our method 27 p.	14.12	98.25	99.29	99.71
Our method 40 p.	13.03	98.25	100.01	99.88
Our one-shot 40 p.	15.07	51.50	99.87	99.85
Designer 40 p.	23.55	55.00	100.15	99.99

in Table 2, where both QoIs related to element failure were consistently lower than the threshold for the models above 10 parameters. Since the incremental approach is able to produce models with lower objective function and higher activation of the constraints, we conclude that it is the most suited for the initial design phase.

5 | Conclusions and Perspectives

In this paper, we presented a new hierarchical parameterization refinement method that leverages ROMs and a series of optimization subproblems. The reparameterization procedure does not rely on specific optimization methods and can be adapted to other use cases. We showcased the benefit of such a technique by extending an existing pipeline for the automated structural optimization of cruise ships with the addition of multi-objective optimization capabilities, a specialized BO procedure, a local search heuristic, and the aforementioned parameterization refinement procedure.

The multi-objective optimization is based on the established NSGA GA, coupled with an infill criterion leveraging the GPR component of the surrogates. The BO procedure has been enhanced with the addition of linear inequality constraints on the incumbent and the VCG, and an ILP-based rounding procedure to overcome the rounding issues due to the discrete nature of the parametric space. The PDS provides a local, black-box refinement procedure to further optimize the surrogates when BO is not able to find new candidates within the computation budget. Finally, the reparameterization module refines the optimization problem formulation by adapting to the emergent behavior of the QoIs. New parameters are constructed by finding a clustering of patches in each parameterized section to separate the responses to thickness changes in terms of yielded and buckled elements. The clusters are determined through the solution of ILPs constructed on the surrogate models' predictions.

The hierarchical nature of the refinement enables efficient use of past expensive high-fidelity results. The enhanced pipeline has been tested on the mass optimization of two test cases, a midship section and a large-scale full ship, for which an initial parameterization was refined both iteratively and at once, and the final results were compared with a baseline given by expert designers. In the simpler case, the pipeline results are comparable to the baseline even though a lower number of parameters was used, suggesting the effectiveness of the approach. In the more complex case, the pipeline was able to achieve a much lower total ship mass than the baseline, with a large gain due to the configurations found by the clustering procedure. The proposed pipeline has proven effective in achieving a ship configuration with a low total mass, while respecting the safety and manufacturing constraints imposed by the designers. The use of this pipeline can help streamline the initial design phase of complex or novel ships, providing good configurations with minimal supervision.

Further directions of development could be the integration of other types of decision variables, such as steel quality, or the addition of different buckling models to support more classification rules. Another extension could be the introduction of fatigue modeling, which would, however, require adaptation of the surrogate construction and evaluation. Although formulated for the optimization of cruise ships, the pipeline could be applied to the structural design phase of other structures where the limiting factor is the validation of static yielding and buckling phenomena. QoIs such as the vertical deflection and VCG could be substituted by other quantities relevant to the new context. The reparameterization module, however, relies only on the assignment of the decision variables to the geometrical elements and can be applied without changes to new problems of the same nature.

Acknowledgments

This work was partially supported by the PNRR (PhD with industries) project "Development of advanced reduced methods of parametric optimization for the structural analysis of cruise vessels," co-sponsored by Fincantieri SpA, by the PNRR NGE project iNEST "Interconnected Nord-Est Innovation Ecosystem," and by the European Research Council Executive Agency by the Proof of Concept project ARGOS, "Advanced Reduced Groupware Online Simulation," P.I. Professor Gianluigi Rozza.

Funding

This work was supported by Italy's PNRR by MUR (Grant No. DM 117/2023).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Endnotes

[1] In this work, "parameterization" denotes the specific assignment of mesh elements to decision variables of an optimization problem. In particular, we are interested in decision variables that specify the thickness of 2D shell elements. The words "decision variable" and "parameter" will be used interchangeably, with a specific instance of thickness values being denoted as "parameter configuration" or simply "configuration."

References

1. M. Tadros, M. Ventura, and C. G. Soares, "Review of Current Regulations, Available Technologies, and Future Trends in the Green Shipping Industry," *Ocean Engineering* 280 (2023): 114670, https://doi.org/10.1016/j.oceaneng.2023.114670.

2. R. Roy, S. Hinduja, and R. Teti, "Recent Advances in Engineering Design Optimisation: Challenges and Future Trends," *CIRP Annals* 57, no. 2 (2008): 697–715, https://doi.org/10.1016/j.cirp.2008.09.007.

3. M. B. Giles and N. A. Pierce, "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion* 65, no. 3/4 (2000): 393–415, https://doi.org/10.1023/A:1011430410075.

4. A. Bayatfar, J. Mishael, R. Warnotte, and P. Rigo, *An Integrated Framework for Ship Structural Optimisation in Contract Design Phase. Annals of Dunărea de Jos*, vol. 42 (University of Galati, 2019), 109–116, https://doi.org/10.35219/AnnUGalShipBuilding.2019.42.15.

5. B. S. Jang, J. D. Kim, T. Y. Park, and S. B. Jeon, "FEA Based Optimization of Semi-Submersible Floater Considering Buckling and Yield Strength," *International Journal of Naval Architecture and Ocean Engineering* 11, no. 1 (2019): 82–96, https://doi.org/10.1016/j.ijnaoe.2018.02.010.

6. J. Raikunen, E. Avi, H. Remes, J. Romanoff, I. Lillemäe-Avi, and A. Niemelä, "Optimisation of Passenger Ship Structures in Concept Design Stage," *Ships and Offshore Structures* 14, no. sup1 (2019): 320–334, https://doi.org/10.1080/17445302.2019.1590947.

7. A. Ivaldi, A. Bayatfar, J. D. Caprace, et al., "Committee IV.2: Design Methods," in *The Society of Naval Architects & Marine Engineer* (SNAME, 2022) D011S001R009.

8. P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, *Eds.Model Order Reduction - 1 - System- and Data-Driven Methods and Algorithms* (De Gruyter, 2021).

9. J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations* (Springer International Publishing, 2016).

10. F. Chinesta, A. Huerta, G. Rozza, and K. Willcox, *Model Reduction Methods* (John Wiley & Sons, Ltd., 2017), 1–36.

11. S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd ed. (Cambridge University Press, 2022).

12. J. Li, X. Du, and J. R. Martins, "Machine Learning in Aerodynamic Shape Optimization," *Progress in Aerospace Science* 134 (2022): 100849, https://doi.org/10.1016/j.paerosci.2022.100849.

13. A. Schrijver, *Theory of Linear and Integer Programming* (John Wiley & Sons, 1998).

14. M. Tezzele, L. Fabris, M. Sidari, M. Sicchiero, and G. Rozza, "A Multifidelity Approach Coupling Parameter Space Reduction and Nonintrusive POD With Application to Structural Optimization of Passenger Ship Hulls," *International Journal for Numerical Methods in Engineering* 124, no. 5 (2023): 1193–1210, https://doi.org/10.1002/nme.7159.

15. C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2005).

16. R. B. Gramacy, *Surrogates - Gaussian Process Modeling, Design, and Optimization for the Applied Sciences* (Chapman & Hall/CRC, 2021).

17. B. Shahriari, K. Swersky, Z. Wang, and R. P. Adams, "Freitas dN. Taking the Human out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE* 104, no. 1 (2016): 148–175, https://doi.org/10.1109/JPROC.2015.2494218.

18. S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms* (Springer, 2008).

19. C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation* 3, no. 1 (1995): 1–16, https://doi.org/10.1162/evco.1995.3.1.1.

20. A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective Evolutionary Algorithms: A Survey of the State of the Art," *Swarm and Evolutionary Computation* 1, no. 1 (2011): 32–49, https://doi.org/10.1016/j.swevo.2011.03.001.

21. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation* 6, no. 2 (2002): 182–197, https://doi.org/10.1109/4235.996017.

22. S. Horne, "MSC/Nastran," in *Finite Element Systems*, ed. C. A. Brebbia (Springer, 1982), 287–294.

23. D. N. Veritas, "Rules for Classification of Ships," 2017.

24. L. N. Trefethen and D. I. Bau, *Numerical Linear Algebra* (Society for Industrial and Applied Mathematics, 1997).

25. R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox, "Projection-Based Model Reduction: Formulations for Physics-Based Machine Learning," *Computers and Fluids* 179 (2019): 704–717, https://doi.org/10.1016/j.compfluid.2018.07.021.

26. N. Demo, M. Tezzele, A. Mola, and G. Rozza, "Hull Shape Design Optimization With Parameter Space and Model Reductions, and Self-Learning Mesh Morphing," *Journal of Marine Science and Engineering* 9, no. 2 (2021): 185, https://doi.org/10.3390/jmse9020185.

27. S. Fresca and A. Manzoni, "POD-DL-ROM: Enhancing Deep Learning-Based Reduced Order Models for Nonlinear Parametrized PDEs by Proper Orthogonal Decomposition," *Computer Methods in Applied Mechanics and Engineering* 388 (2022): 114181, https://doi.org/10.1016/j.cma.2021.114181.

28. M. Guo and J. S. Hesthaven, "Reduced Order Modeling for Nonlinear Structural Analysis Using Gaussian Process Regression," *Computer Methods in Applied Mechanics and Engineering* 341 (2018): 807–826, https://doi.org/10.1016/j.cma.2018.07.017.

29. M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization* (MIT Press, 2019).

30. C. Czech, M. Lesjak, C. Bach, and F. Duddeck, "Data-Driven Models for Crashworthiness Optimisation: Intrusive and Non-Intrusive Model Order Reduction Techniques," *Structural and Multidisciplinary Optimization* 65, no. 7 (2022): 190, https://doi.org/10.1007/s00158-022-03282-1.

31. L. Cicci, S. Fresca, M. Guo, A. Manzoni, and P. Zunino, "Uncertainty Quantification for Nonlinear Solid Mechanics Using Reduced Order Models With Gaussian Process Regression," *Computers & Mathematics With Applications* 149 (2023): 1–23, https://doi.org/10.1016/j.camwa.2023.08.016.

32. Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation* 11, no. 6 (2007): 712–731, https://doi.org/10.1109/TEVC.2007.892759.

33. J. Knowles, "ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Transactions on Evolutionary Computation* 10, no. 1 (2006): 50–66, https://doi.org/10.1109/TEVC.2005.851274.

34. H. Ohno, "Analysis of Multi-Objective Bayesian Optimization Using Random Scalarizations for Correlated Observations," *SN Computer Science* 2, no. 3 (2021): 214, https://doi.org/10.1007/s42979-021-00505-y.

35. S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Multi-Objective Bayesian Optimization Over High-Dimensional Search Spaces," in *Proceedings of Machine Learning Research. PMLR* (PMLR, 2022).

36. M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Transactions on Evolutionary Computation* 10, no. 4 (2006): 421–439, https://doi.org/10.1109/TEVC.2005.859463.

37. K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation* 18, no. 4 (2014): 577–601, https://doi.org/10.1109/TEVC.2013.2281535.

38. S. Seo, M. Wallat, T. Graepel, and K. Obermayer, *Gaussian Process Regression: Active Data Selection and Test Point Rejection*, vol. 3 (IEEE, 2000), 241–246.

39. D. J. C. MacKay, "Information-Based Objective Functions for Active Data Selection," *Neural Computation* 4, no. 4 (1992): 590–604, https://doi.org/10.1162/neco.1992.4.4.590.

40. D. A. Cohn, "Neural Network Exploration Using Optimal Experiment Design," *Neural Networks* 9, no. 6 (1996): 1071–1083, https://doi.org/10.1016/0893-6080(95)00137-9.

41. D. Hernandez-Lobato, J. Hernandez-Lobato, A. Shah, and R. Adams, "Predictive Entropy Search for Multi-objective Bayesian Optimization," in *Proceedings of The 33rd International Conference on Machine Learning. 48 of Proceedings of Machine Learning Research*, ed. M. F. Balcan and K. Q. Weinberger (PMLR, 2016), 1492–1501.

42. S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value Entropy Search for Multi-Objective Bayesian Optimization," in *Advances in Neural Information Processing Systems*, vol. 32, ed. H. Wallach, H. Larochelle, A. Beygelzimer, D. F. Alché-Buc, E. Fox, and R. Garnett (NeurIPS Foundation. Curran Associates, Inc, 2019).

43. S. Suzuki, S. Takeno, T. Tamura, K. Shitara, and M. Karasuyama, "Multi-objective Bayesian Optimization using Pareto-frontier Entropy," in *Proceedings of the 37th International Conference on Machine Learning. 119 of Proceedings of Machine Learning Research*, ed. H. C. Daumé and A. Singh (PMLR, 2020), 9279–9288.

44. N. Srinivas, A. Krause, S. Kakade, and M. W. Seeger, "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design," in *ICML'10. International Machine Learning Society* (Omnipress, 2010), 1015–1022.

45. D. Zhan and H. Xing, "Expected Improvement for Expensive Optimization: A Review," *Journal of Global Optimization* 78, no. 3 (2020): 507–544, https://doi.org/10.1007/s10898-020-00923-x.

46. D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization* 13 (1998): 455–492, https://doi.org/10.1023/A:1008306431147.

47. J. R. Gardner, M. J. Kusner, Z. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian Optimization With Inequality Constraints," in *ICML'14. International Machine Learning Society* (JMLR.org, 2014), II–937–II–945.

48. V. Picheny, R. B. Gramacy, S. M. Wild, and S. L. Digabel, "Bayesian Optimization Under Mixed Constraints With a Slack-Variable Augmented Lagrangian," in *NIPS'16. NeurIPS Foundation* (Curran Associates Inc, 2016), 1443–1451.

49. A. Paleyes, M. Pullin, M. Mahsereci, et al., *Emulation of Physical Processes With Emukit* (NeurIPS Foundation, 2019).

50. A. Paleyes, M. Mahsereci, and N. D. Lawrence, "Emukit: A Python Toolkit for Decision Making Under Uncertainty. Proceedings of the Python in Science Conference," (2023).

51. P. Virtanen, R. Gommers, T. E. Oliphant, et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods* 17 (2020): 261–272, https://doi.org/10.1038/s41592-019-0686-2.

52. R. H. Byrd, M. E. Hribar, and J. Nocedal, "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM Journal on Optimization* 9, no. 4 (1999): 877–900, https://doi.org/10.1137/S1052623497325107.

53. E. C. Garrido-Merchán and D. Hernández-Lobato, "Dealing With Categorical and Integer-Valued Variables in Bayesian Optimization With Gaussian Processes," *Neurocomputing* 380 (2020): 20–35, https://doi.org/10.1016/j.neucom.2019.11.004.

54. C. Oh, J. M. Tomczak, E. Gavves, and M. Welling, "Combinatorial Bayesian Optimization using the Graph Cartesian Product," in *NeurIPS Foundation* (Curran Associates Inc, 2019).

55. J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems*, vol. 24, ed. J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (NeurIPS Foundation. Curran Associates Inc, 2011).

56. F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential Model-Based Optimization for General Algorithm Configuration," in *Learning and Intelligent Optimization*. Lecture Notes in Computer Science, vol. 6683, ed. C. A. C. Coello (Springer Berlin Heidelberg, 2011), 507–523.

57. R. Baptista and M. Poloczek, "Bayesian Optimization of Combinatorial Structures," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, Dy, J. and Krause, A., ed. (PMLR, 2018), 462–471.

58. E. Daxberger, A. Makarova, M. Turchetta, and A. Krause, "Mixed-Variable Bayesian Optimization," in *International Joint Conferences on Artificial Intelligence Organization* (International Joint Conferences on Artificial Intelligence Organization, 2020), 2633–2639.

59. J. Forrest, T. Ralphs, S. Vigerske, et al., "coin-or/Cbc: Release Releases/2.10.11," (2023).

60. H. G. Santos and T. Toffolo, "Mixed Integer Linear Programming With python. COINOR Computational Infrastructure for Operations Research," (2020).

61. M. A. Benaimeche, J. Yvonnet, B. Bary, and Q. He, "A k-Means Clustering Machine Learning-Based Multiscale Method for Anelastic Heterogeneous Structures With Internal Variables," *International Journal for Numerical Methods in Engineering* 123, no. 9 (2022): 2012–2041, https://doi.org/10.1002/nme.6925.

62. D. A. Spielman and S. H. Teng, "Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time," *Journal of the Association for Computing Machinery* 51, no. 3 (2004): 385–463, https://doi.org/10.1145/990308.990310.

63. T. Achterberg and R. Wunderling, "Mixed Integer Programming: Analyzing 12 Years of Progress," in *Facets of Combinatorial Optimization*, ed. M. Jünger and G. Reinelt (Springer, 2013), 449–481.

Appendix A

Optimization Pipeline Flowchart

The complete flowchart for the optimization pipeline described in Section 2.2 is represented in Figure A1.

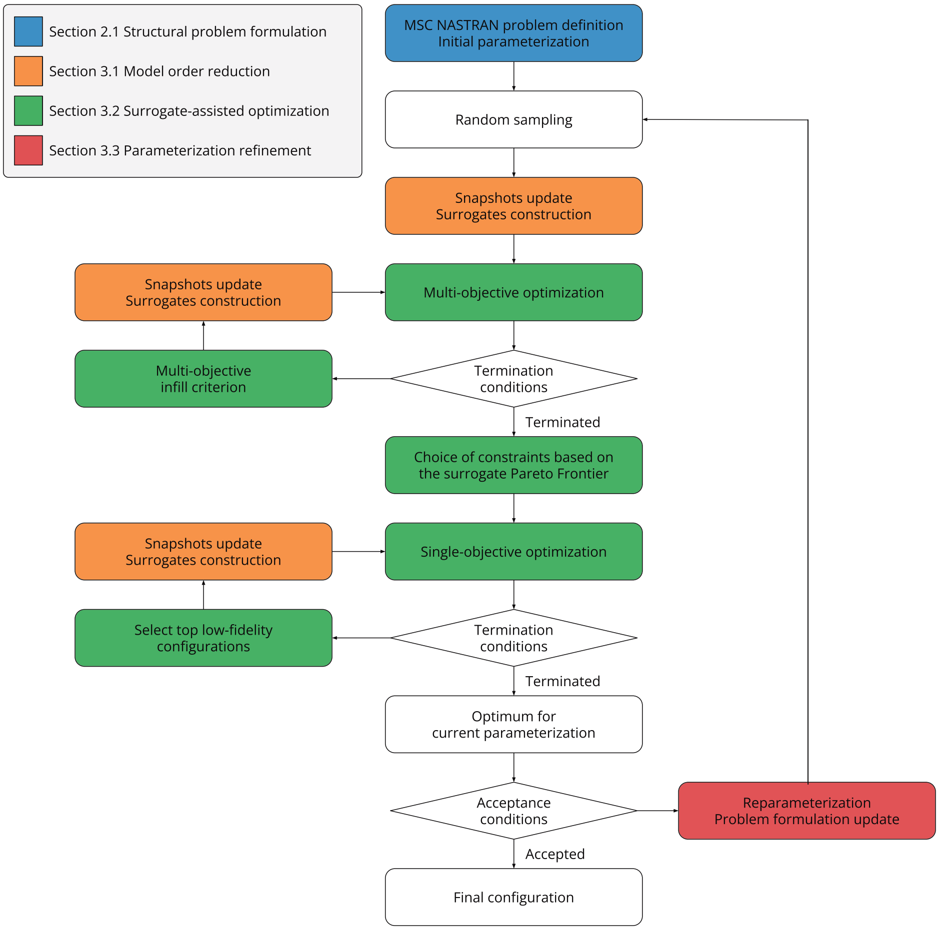

FIGURE A1 | Flowchart for the optimization pipeline. Process nodes are color-coded with the relevant sections in the paper. In decision nodes, the user evaluates the trade-offs between the optimization results, the computational effort and the project's resource constraints.

Appendix B

Reparameterization Schema

The reparameterization procedure from Section 3.3 is represented in Figure B1, applied to the midship section. In this case, the initial model had 5 parameterized sections and the procedure found 2 clusters in each, generating 5 additional parameterized sections for a total of 10.

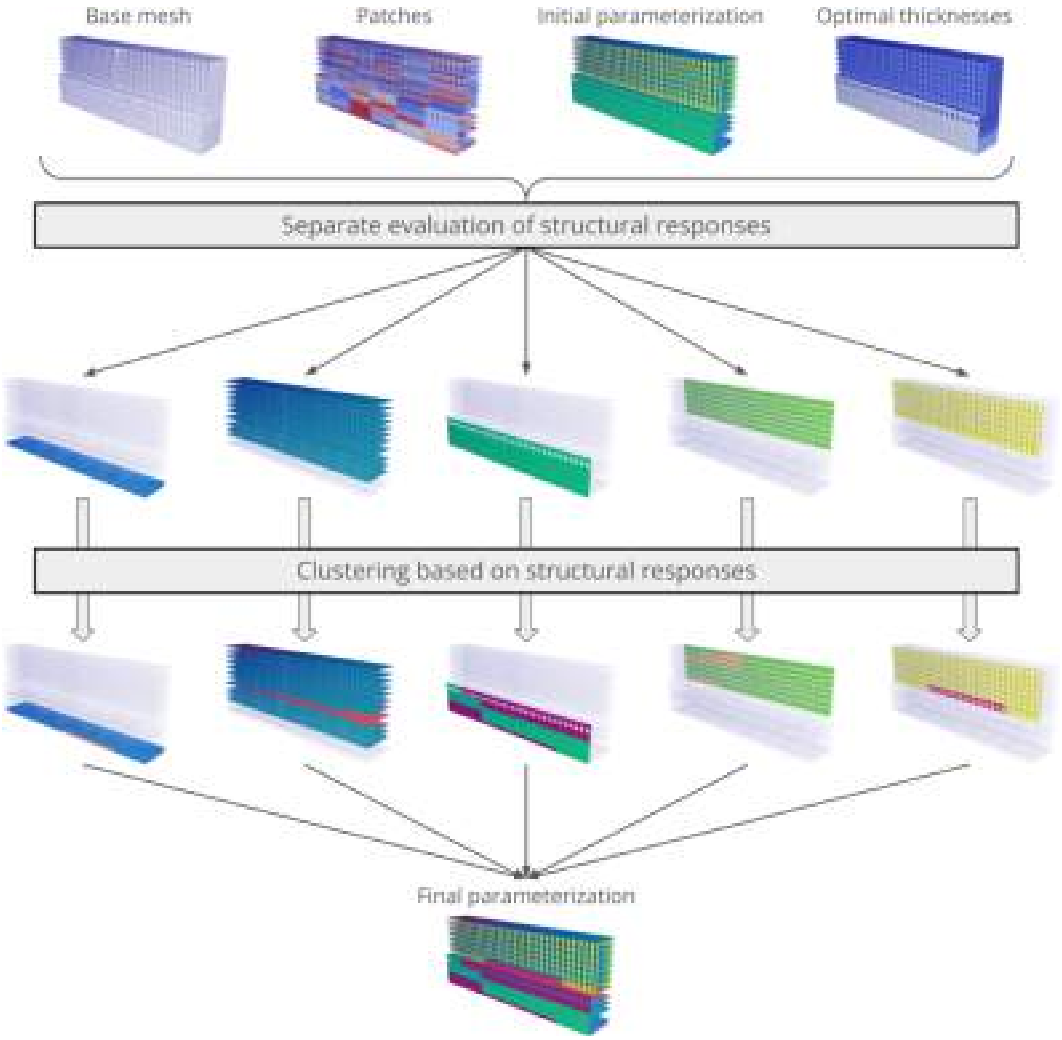

FIGURE B1 | The reparameterization process requires the computational mesh, whose elements are grouped in patches and assigned to parameters, and a reference thickness configuration. The patches of each parameterized section are clustered according to the structural responses, and additional decision variables are created following the clusters. The new parameterization for the model can then better control the QoIs of the simulation, enabling the achievement of better optima at the cost of a more complex optimization problem.

Appendix C

Evaluation of the Errors in the Surrogate Models

Surrogate models are used to quickly select promising configurations for the minimization of the QoIs. Due to the expensive validation of the results using the high-fidelity solver, the database of configurations, from which the ROMs are constructed, mostly comprises the results from multi- and single-objective optimizations, as selected by their infill criteria. The only exceptions come from the initial random sampling, and from the reparameterization steps, where the larger domain is again randomly sampled before progressing to the optimization.

We evaluate the choice of the truncation rank used in the POD, and its update after the reparameterization. Automatization of these steps will make the pipeline more user-friendly and provide smoother integration with existing workflows. We report on the behavior of the POD singular values and the prediction errors for different combinations of truncation rank and high-fidelity database sizes. In Section C we analyze the models based on the midship section, while Section C presents the full ship results.

Midship Section

Figure C1 shows the evolution of the high-fidelity optimum for the midship section, for the iterative reparameterization approach described in Section 4.1. The plot begins at the 43rd sample, as the earlier ones are significantly higher than the final optimum. Multi-objective optimization provides the largest component of the high-fidelity database, with configurations spread throughout the domain, while single-objective optimization limits configurations close to the latest optimum.

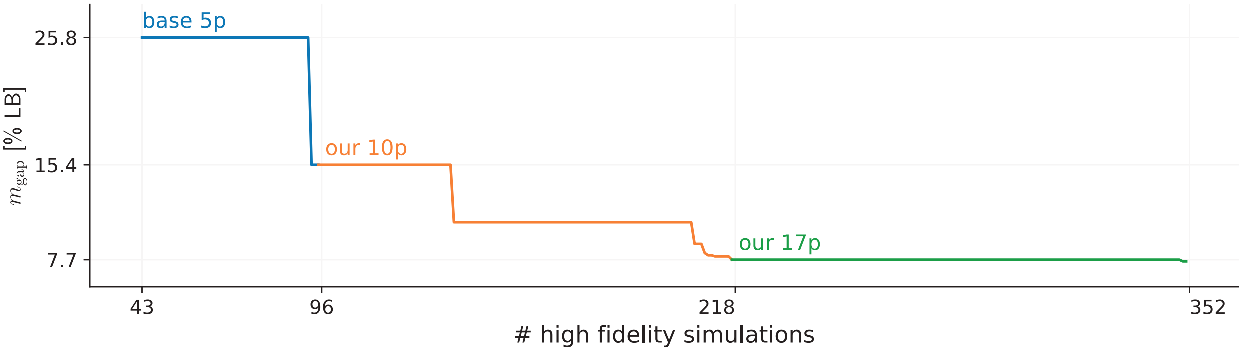

FIGURE C1 | Evolution of the high-fidelity optimal configuration for the midship section, under the iterative reparameterization approach. For a comparison with different settings see Figure 13.

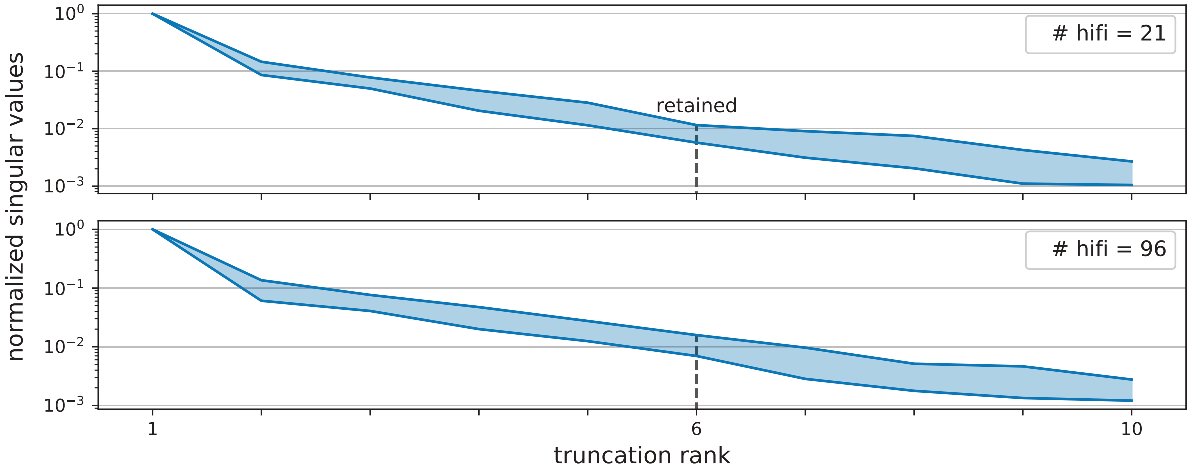

FIGURE C2 | Evolution of the minimum and maximum for the normalized singular values decay of the stress tensor components, for the midship section with 5 parameters. Values are reported for the high-fidelity database before and after optimization.

The choice of truncation rank for the POD happens after the initial random sampling. Figure C2 shows the evolution of the singular values decay for the stress tensor components, for different database sizes. The databases considered are the initial sampling (the initial configuration and 20 random samples) and the final one, after single-objective optimization. For the initial sampling, the normalized singular values from rank 7 and above fall below the 10^{-2} threshold and are thus discarded. As new high-fidelity samples are acquired, the largest 7-th singular value never exceeds the 10^{-2} threshold, confirming that the truncation choice was robust.

Figure C3 shows the behavior of the 5-fold cross-validation error on the number of yielded and buckled elements, as a fraction of the critical threshold (200 for the number of yielded elements, 4000 for the buckled elements). We also show the results for a database approximately in the middle of the procedure, during the multi-objective optimization phase. The initial database does not show a clear decrease of prediction errors as the truncation rank grows, while during the middle of the pipeline, the surrogates achieve good predictive performances. With the largest database, the surrogates show little difference between truncation ranks 6 and 7. The choice of truncation rank 6 appears to be appropriate for the initial parameterization of the midship section.

Figures C4 and C5 show the same information discussed before, but for the model after its reparameterization to 10 parameters. The first database considered contains the previous 96 samples, and 20 random samples from the enlarged domain. The truncation rank was automatically increased from 6 to 11, and the first POD of the stress tensors shows how the largest normalized singular values reach the 10^{-2} threshold between ranks 9 and 11. Further enrichments of the high-fidelity database determine a slower decay, but the truncation rank at 11 still discards the modes with a normalized singular value below 10^{-2}. As for the prediction error of yielded and buckled elements, the first surrogates constructed on the enlarged domain show an increase in the maximum error values compared to the latest surrogates using 5 parameters. The truncation rank 11 is compared to 6, 16, and 21, showing that retaining a larger number of modes does not provide substantial gains in accuracy, and suggesting instead an increase in prediction errors due to outliers. Performances are comparable to the 5-parameters models in term of median values.

With the second reparameterization of the model, the new domain has 17 dimensions and the truncation rank is increased to 18. Again, the initial database contains the 218 configurations from the previous optimization, and 20 samples from the new domain. This time,

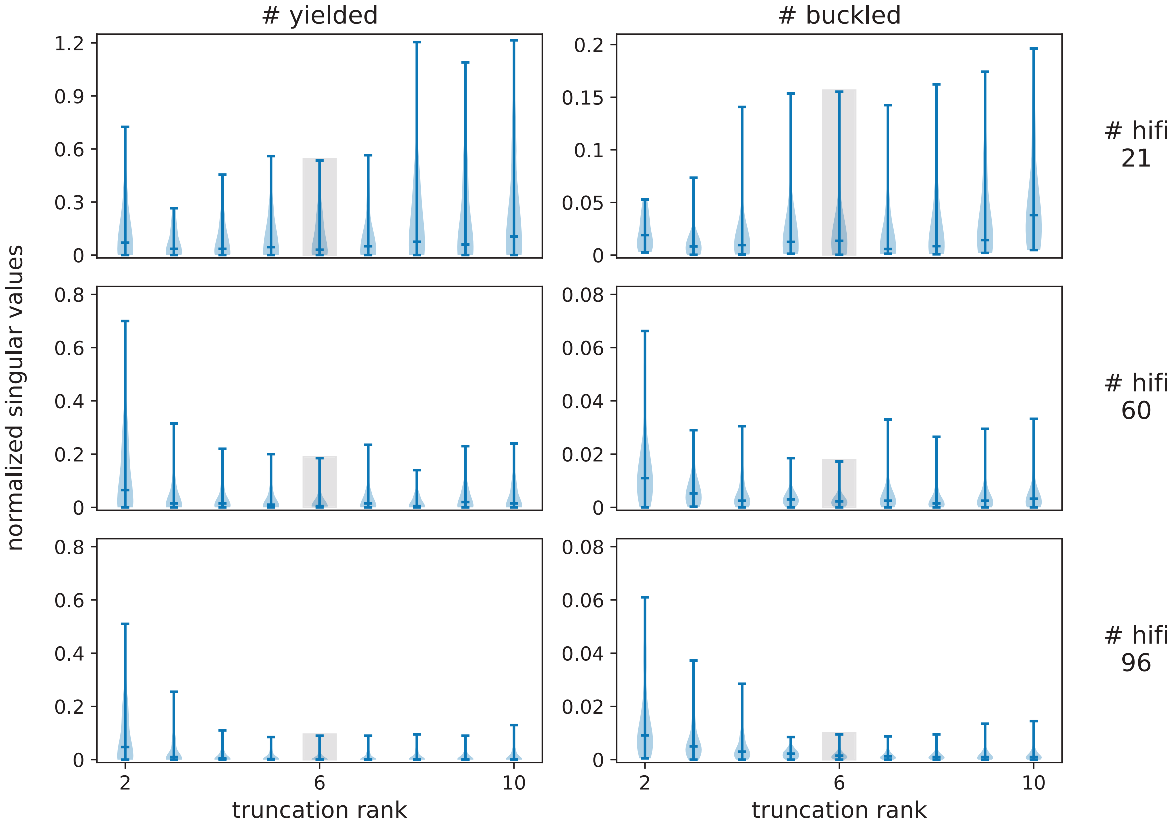

FIGURE C3 | Evolution of the 5-fold cross-validation error distribution for the midship section with 5 parameters, for different sizes of the high-fidelity database. For the number of yielded and buckled elements, the difference between the predicted and true values is divided by the critical threshold. The plots highlight the median error and the gray box identifies the chosen truncation rank.

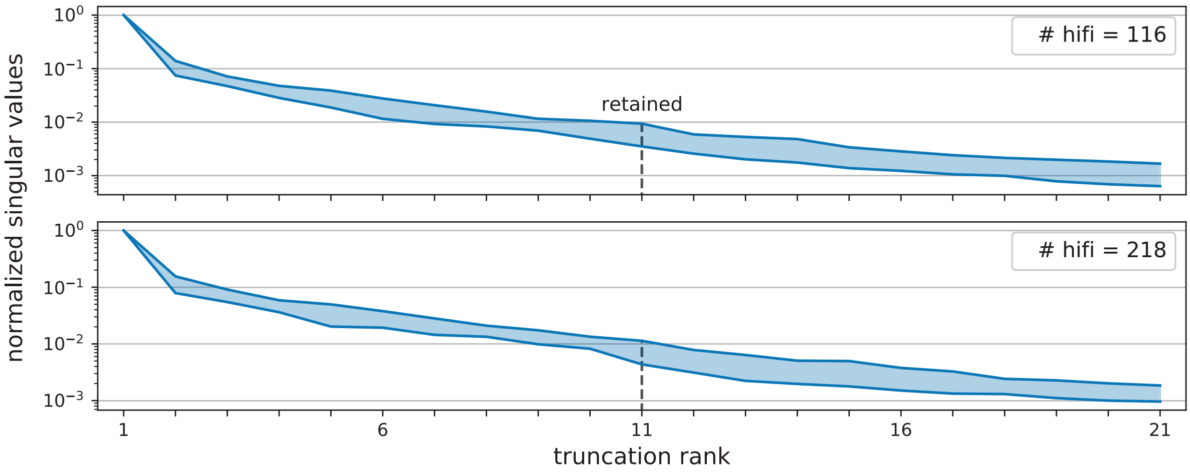

FIGURE C4 | Evolution of the minimum and maximum for the normalized singular values decay of the stress tensor components, for the midship section with 10 parameters. Values are reported for the high-fidelity database before and after optimization.

the normalized singular values decay in Figure C6 crosses the 10^{-2} threshold between ranks 14 and 15, and further exploration of the domain only moves it between ranks 16 and 17. In this case, the automatic update of the truncation rank was conservative. The behavior of the error on the number of yielded elements does not show a marked dependence on the truncation ranks tested, as shown in Figure C7. As for the error on the number of buckled elements, the largest high-fidelity database shows only a marginally lower median error from rank 16 and above, compared to rank 11.

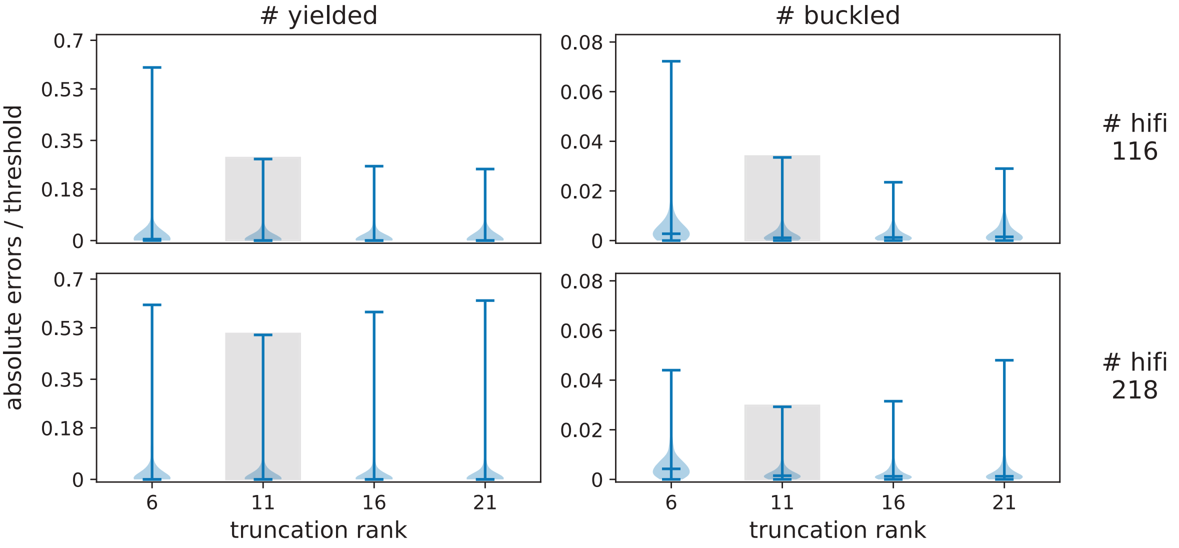

FIGURE C5 | Evolution of the 5-fold cross-validation error distribution for the midship section with 10 parameters. Values are reported for the high-fidelity database before and after optimization. The gray box identifies the chosen truncation rank.

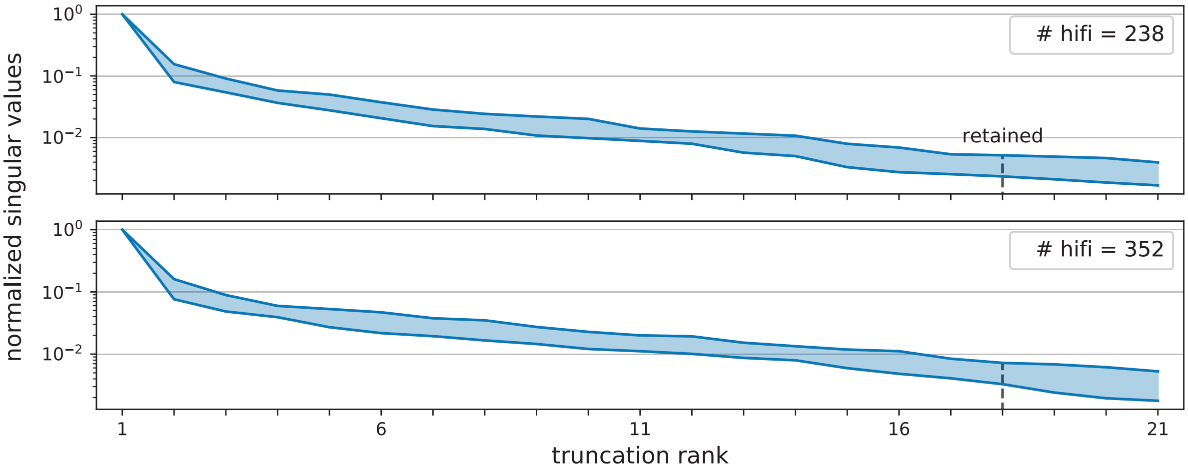

FIGURE C6 | Evolution of the minimum and maximum for the normalized singular values decay of the stress tensor components, for the midship section with 17 parameters. Values are reported for the high-fidelity database before and after optimization.

The results from the evaluation validate the choice of 10^{-2} as the threshold for the truncation rank in the POD, at the start of the optimization pipeline. This value is unusually high compared to the common practice, but as shown by the results for the prediction errors, choosing a higher truncation rank seems to decrease, rather than increase, the accuracy of the ROMs for evaluating the QoIs. A possible explanation is due to the combination of an increase in the complexity of performing regression on a larger latent space, and the non-linearity of the post-processing step for computing the QoIs, especially the buckling state of the elements. Indeed, the singular values decay does not provide information on the variability of the coefficients across the domain nor in the areas close to the best-performing configurations, so reconstruction errors in latent space could lead to larger effects on the QoIs.

Full Ship

The evolution of the high-fidelity optimum, during the iterative reparameterization of the full ship model, is depicted in Figure C8. The plot begins at the 50th sample, as the earlier ones are significantly higher than the final optimum.

Figure C9 shows the evolution of the singular values decay for the model with the initial parameterization. With the initial sampling of 21 configurations, the normalized singular values from rank 14 and above fall below the 10^{-2} threshold, but the truncation rank was set to 16 as a precaution against the possible complexity arising from the larger number of parameters. As the optimizations explores the design space, this proved to be too conservative, and the singular values remained below the 10^{-2} threshold for ranks 14 and above. Figure C10 shows the evolution of the errors on the yielded and buckled elements. The maximum errors on the yielded elements are much larger than those measured on the buckled elements, but the addition of high-fidelity configurations to the database determines

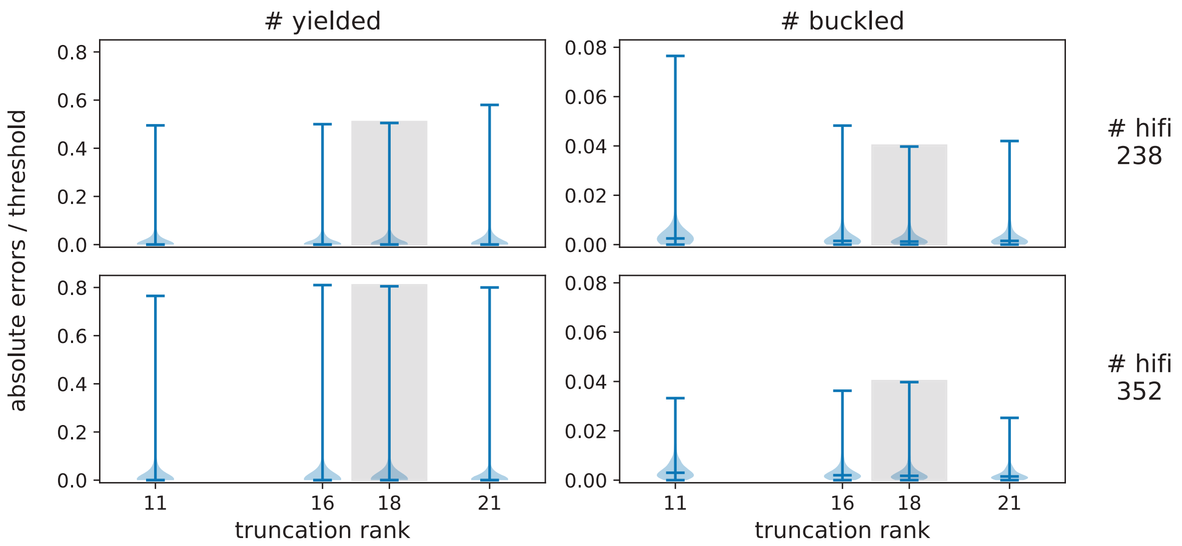

FIGURE C7 | Evolution of the 5-fold cross-validation error distribution for the midship section with 17 parameters. Values are reported for the high-fidelity database before and after optimization. The gray box identifies the chosen truncation rank.

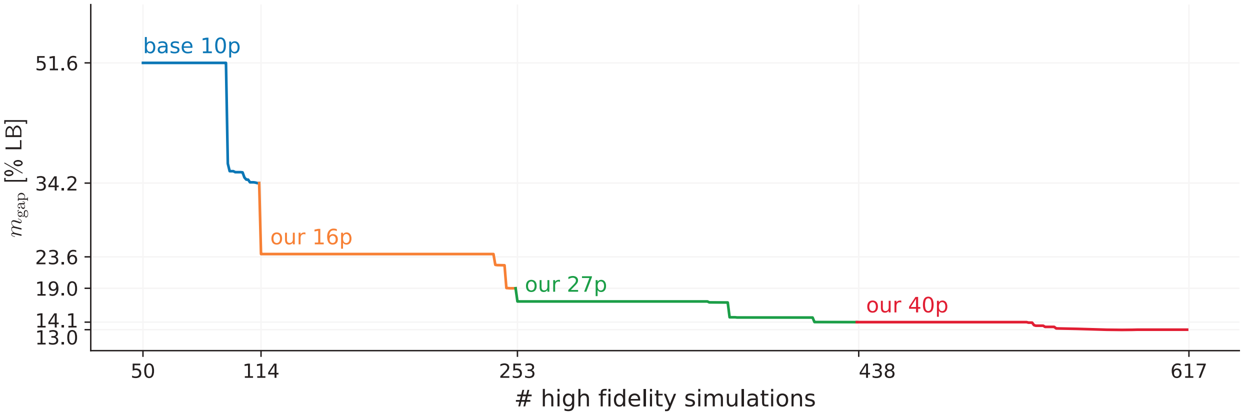

FIGURE C8 | Evolution of the high-fidelity optimal configuration for the full ship, under the iterative reparameterization approach. For a comparison with different settings see Figure 16.

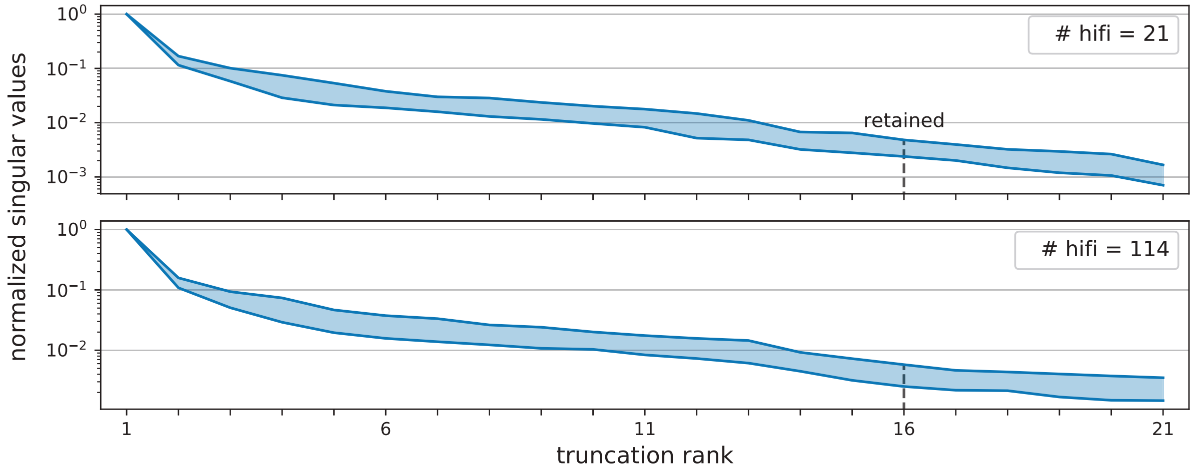

FIGURE C9 | Evolution of the minimum and maximum for the normalized singular values decay of the stress tensor components, for the full ship with 10 parameters. Values are reported for the high-fidelity database before and after optimization.

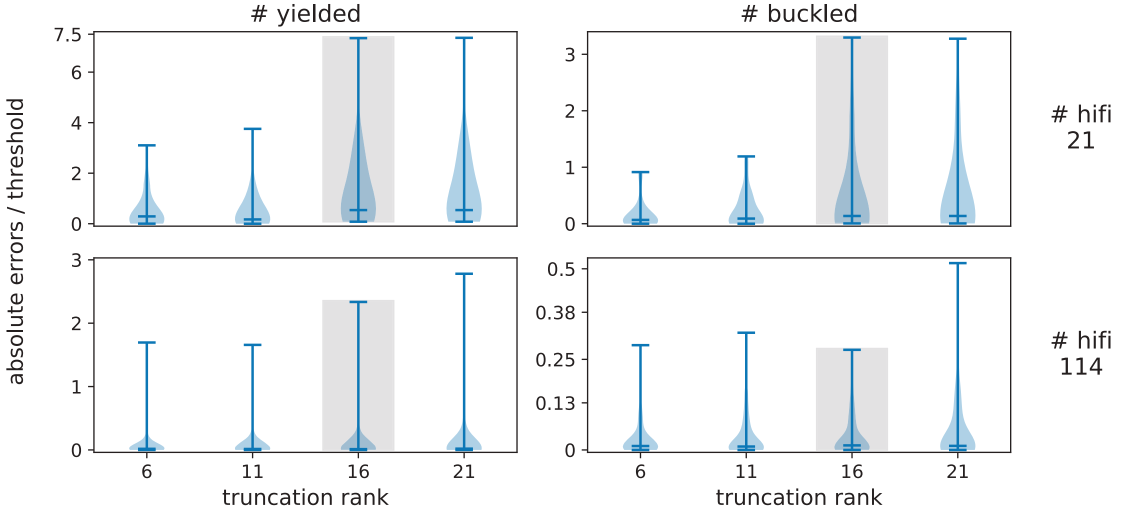

FIGURE C10 | Evolution of the 5-fold cross-validation error distribution for the full ship with 10 parameters. Values are reported for the high-fidelity database before and after optimization. The gray box identifies the chosen truncation rank.

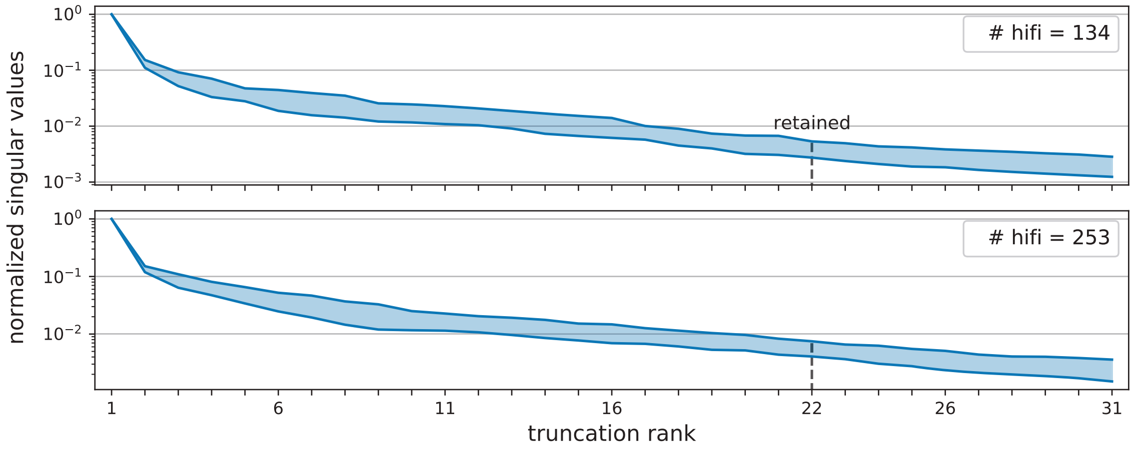

FIGURE C11 | Evolution of the minimum and maximum for the normalized singular values decay of the stress tensor components, for the full ship with 16 parameters. Values are reported for the high-fidelity database before and after optimization.

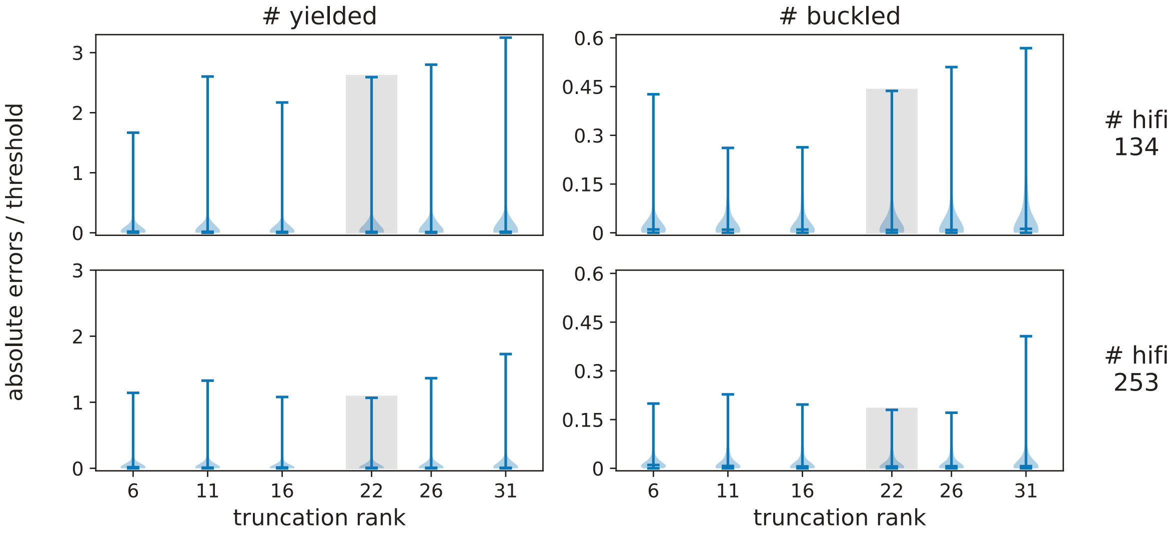

FIGURE C12 | Evolution of the 5-fold cross-validation error distribution for the full ship with 16 parameters. Values are reported for the high-fidelity database before and after optimization. The gray box identifies the chosen truncation rank.

a substantial concentration of the errors close to zero. Unexpectedly, the error distributions corresponding to truncation rank 6 appear comparable to those obtained using higher truncation ranks. This can be linked to the effect of the selection bias from the optimization procedures: 4 of the original 10 parameters showed very little influence on the QoIs, and neither multi- nor single-objective optimization moved them beyond their minimum values. This results in the high-fidelity database underutilizing 4 out of 10 dimensions of the parametric domain, and as the number of validated configurations grows, the prediction errors of the ROMs reflect that the actual rank is about 4 positions lower than the number of parameters.

After the first reparameterization, the truncation rank is again increased by the number of new decision variables. Of the 10 parameters, 6 were split in two, with the remaining 4 deemed not valuable for further refinement. Again, the automatic adjustment of the truncation rank to 22 proves sufficient with respect to the 10^{-2} threshold on the normalized singular values. This situation is depicted in Figure C11. As for the errors on the QoIs, the same trend as for the previous parameterization is seen, with the error distributions being very similar for different truncation ranks, but showing a concentration close to zero as the high-fidelity database size increases (Figure C12).