Review

A Comparative Analysis of Optimization Algorithms for Finite Element Model Updating on Numerical and Experimental Benchmarks

Davide Raviolo [1] , Marco Civera [2,*] and Luca Zanotti Fragonara [3]

1   Department of Structural Engineering, Norwegian University of Science and Technology (NTNU), 7491 Trondheim, Norway; davide.raviolo@ntnu.no
2   Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, 10129 Turin, Italy
3   School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 0AL, UK; l.zanottifragonara@cranfield.ac.uk
*   Correspondence: marco.civera@polito.it

Abstract: Finite Element Model Updating (FEMU) is a common approach to model-based Non-Destructive Evaluation (NDE) and Structural Health Monitoring (SHM) of civil structures and infrastructures. Its application can be further utilized to produce effective digital twins of a permanently monitored structure. The FEMU concept, simple yet effective, involves calibrating and/or updating a numerical model based on the recorded dynamic response of the target system. This enables to indirectly estimate its material parameters, thus providing insight into its mass and stiffness distribution. In turn, this can be used to localize structural changes that may be induced by damage occurrence. However, several algorithms exist in the scientific literature for FEMU purposes. This study benchmarks three well-established global optimization techniques—namely, Generalized Pattern Search, Simulated Annealing, and a Genetic Algorithm application—against a proposed Bayesian sampling optimization algorithm. Although Bayesian optimization is a powerful yet efficient global optimization technique, especially suitable for expensive functions, it is seldom applied to model updating problems. The comparison is performed on numerical and experimental datasets based on one metallic truss structure built in the facilities of Cranfield University. The Bayesian sampling procedure showed high computational accuracy and efficiency, with a runtime of approximately half that of the alternative optimization strategies.

Keywords: finite element model updating; model calibration; digital twin; generalized pattern search; simulated annealing; genetic algorithm; Bayesian expected improvement

1. Introduction

For civil engineering purposes, predictive models are of extreme importance for asset management, lifecycle assessment, and condition-based maintenance. This is particularly true for detailed numerical (Finite Element) models, as their continuous updating enables many other uses of what is generally referred to as a digital twin of the structure/infrastructure of interest, integrating sensed information in an effective replica of the target system [1].

In this context, Finite Element Model Updating (FEMU) strategies are of major interest to ensure accurate and computationally efficient updates, both for minor adjustments after scheduled periodical checks or for major revisions after large (planned or accidental) structural changes, such as retrofitting and seismic events.

These FEMU strategies can be classified as either direct or indirect. The latter techniques are also known as sensitivity-based approaches [2]; the procedure followed in this work falls into this group. For a more detailed discussion, the reader can refer to the textbook of Friswell and Mottershead [3].

As the name suggests, direct methods iteratively minimize the difference between the current and the known matrices of stiffness and damping. Inverse methods, conversely, try to approach the unknown stiffness and matrix distributions by minimizing the difference in the model predictions. This is generally achieved by employing modal parameters and comparing the results of dynamic simulations with the estimates from in situ dynamic monitoring or survey results. This process not only enables calibrating a novel FE model on experimental data; it also allows updating an existing one, potentially calibrated on the structure's pristine response, thus estimating the damage-induced stiffness reduction [4].

The pursuit of model updating encompasses a variety of strategies, contingent upon the ultimate objective of the updating procedure—whether it involves model calibration, identification of system properties, damage detection, etc. These strategies extend to the estimation of parameters associated with stiffness, mass, loads, boundary conditions, and internal element connections, as detailed in [5]. Naturally, the selection of updating strategies and the specific parameters subject to estimation depend on the typology of the structural system under analysis. For illustrative examples of FEMU strategies applied to civil structures and infrastructures, readers are referred to Girardi et al. [6], Ereiz et al. [7], and Nicoletti et al. [8]. Additionally, Arezzo et al. [9] provide insights into FEMU techniques as applied to historical buildings and heritage structures. Furthermore, refs. [10,11] illustrate how model updating of steel trusses can be effectively pursued with static-parameter-identification methods, as an alternative to the somewhat more popular methods based on the dynamic response of the structure. In this framework, ref. [12] makes use of a stiffness separation method to reduce the dimensionality of the problem, thereby improving the convergence rates of the optimizers while preserving the accuracy of the solution.

All optimization methods present a certain trade-off between exploration (i.e., an algorithm's predilection for sampling from areas of high uncertainty) and exploitation (a tendency to sample from areas likely to offer improvement over the current best observation). In the current state of the art, there is no consensus on the most efficient approach to these aims in the context of FEMU. Hence, this research work aims to compare four alternatives (three well-known approaches and a more recent proposal) in relation to a benchmark problem. As shown in the results, Bayesian Sampling Optimization guarantees high accuracy with noticeably lower computational costs.

The remainder of this paper is organized as follows. Section 2 describes the four algorithms benchmarked in this comparative study. Section 3 introduces and details the first case study: a large experimental setup (specifically, an Optical Test System) developed at Cranfield University, which required low transmissibility (i.e., strong vibrational insulation) from ground ambient vibrations for precise measurement. Section 4 reports the results. Finally, Section 5 draws the conclusions based on this work.

2. Benchmarked Optimization Algorithms and Performance Metrics

Four alternative global optimization procedures are considered for this comparative study. These will be detailed in their dedicated subsections; in order, they are (1) Generalized Pattern Search—GPS [13], (2) Simulated Annealing—SA [14], (3) Genetic Algorithm—GA [15], and (4) Bayesian Sampling Optimization—BO [16].

The main conceptual aspects of these techniques are briefly reported below. Importantly, these global optimization algorithms are highly susceptible to specific design choices and initial parameters' values. Hence, the key implementation details are reported as well.

For a fair comparison, the same single objective function was minimized. This is described in Section 2.5. The performance metrics used for the four methodologies are outlined in Section 2.6.

2.1. Generalized Pattern Search (GPS)

The GPS algorithm is, arguably, the simplest and least sophisticated algorithm among the four candidates. It is a direct-search non-probabilistic technique that does not rely

on the computation of the gradients. The first formulation of GPS was given by Hooke and Jeeves in 1961 [17]. The key element of a GPS algorithm is the pattern, by which the algorithm individuates a set of points (called mesh), surrounding the current position, where the objective is evaluated at each step (called the polling phase).

The details of the algorithm implementation used here are as follows:

1. The search bounds of the input parameters are linearly scaled to the interval [0, 100]. This is needed since the employed mesh size is equal in all dimensions.
2. At every successful poll, the mesh size is doubled. Conversely, it is halved after any unsuccessful poll.
3. The algorithm is stopped when the maximum number of objective function evaluations is reached.

2.2. Simulated Annealing (SA)

SA algorithms are often used to find the global minimum in optimization problems characterized by many local minima. Applied to minimization problems and popularized by [18], the concept behind these heuristic techniques comes from an analogy to the annealing treatment of physical materials (such as metals). A very early mention of this method can be found in [19], where an algorithm to simulate the annealing process at the computational level was proposed. According to the SA technique, the input parameter values represent the state of the system, the objective (or fitness) function acts as the energy function, and the parameter that controls the optimization (annealing) process can be seen as a temperature variable.

To initialize the procedure, an initial parameter input vector (initial system state) is randomly generated, and its fitness value (objective) is computed. A new trial point is then chosen by the neighborhood function. This describes the distance of the new point from the current one with a scale dependent on the current temperature. Finally, if the newly sampled point is fitter than the current one, it is accepted and becomes the next point; if the new point is less fit, it is accepted with a probability given by the acceptance function. This function is sensitive to the difference between the new and the previous fitness values as well as to the temperature parameter. The temperature is lowered at each iteration according to a given cooling schedule.

The dependency on the specific implementation details is particularly relevant for SA, as numerous neighborhood functions and cooling schedules can be adopted. While the cooling rate should be sufficiently slow to approach the condition of thermodynamic equilibrium (i.e., find the global optimum), an excessively low cooling rate leads to very slow convergence. Ingber [20] provides additional information on the SA algorithm as well as a review of the many shades SA can assume when following different implementation approaches.

For this research, two different strategies were initially envisioned. The first consisted of using a high cooling rate while reannealing several times during the optimization process. The second made use of a much lower cooling rate, but reannealing was not permitted. The two different implementations are detailed as follows:

Strategy 1:

1. The initial temperature T_0 is set at 100.
2. The temperature gradually decreases at each iteration according to the (exponential) cooling schedule $T = T_0 \cdot 0.95^k$, where k is equal to the iteration number.
3. The reannealing function selects the next point in a random direction, with a step length equal to the current temperature T.
4. If a sampled point is less fit, it is accepted according to the acceptance function $\frac{1}{1 + exp\left(\frac{\Delta}{max(T)}\right)}$, where Δ is the difference between the objective values.
5. Reannealing occurs every 100 consecutively accepted sampled points.

Strategy 2:

1. The initial temperature T_0 is set at 50.
2. The temperature gradually decreases at each iteration according to the (linear) cooling schedule $T = T_0/k$, where k is (as before) the iteration number.
3. The reannealing function selects the next point in a random direction, with a step length equal to the current temperature T.
4. If a sampled point is less fit, it is accepted according to the same acceptance function of Strategy 1, $\frac{1}{1+exp\left(\frac{\Delta}{max(T)}\right)}$.
5. Reannealing never occurs.

For both strategies (and similarly to GPS), the optimization bounds of the input parameters are linearly scaled, in this case, according to the interval [0, 1]. It follows from the above definitions that for the first 90 iterations when using the first strategy and for the first 50 iterations when using the second strategy, the objective is randomly sampled within the optimization bounds. Hence, the choice of the initialization point has no significant influence on the final results.

It was verified that Strategy 2 consistently outperformed Strategy 1 in all aspects; therefore, only the results of the former will be reported hereinafter.

2.3. Genetic Algorithm (GA)

Genetic Algorithm is quite a comprehensive term that can refer to many related yet sensibly different algorithms, ranging from single to multi-objective optimization (see, e.g., [21]). In the most general terms, GA—as described by Holland in his seminal work [15]—seeks the best solution among a population of solutions. This is achieved by applying the principles of evolutionary biology, under the hypothesis that individuals who have a genetic advantage over others have higher chances of breeding successfully. During a cycle of generations, new individuals with enhanced genetic makeup are produced through selection and recombination operations. As these advantaged individuals spread their features over the entire population, this gradually leads to overall improved fitness. Elitist GA approaches, such as the procedure implemented here, also consider a small portion of the best candidate in the current population to be transferred to the next generation without undergoing any change, to slightly encourage exploitation over exploration of the search space.

The following aspects characterize the GA implementation used in this study:

1. Compliance with optimization bounds is enforced by ensuring that each individual is generated (though proper crossover and mutation operators) within the given constraints.
2. The initial population, necessary to initialize the algorithm, consists of 50 randomly chosen points (the initial population size is therefore equal to 10 times the number of dimensions).
3. The crossover fraction is set to 0.8.
4. The elite size is set at 5% of the population size.
5. The mutation fraction varies dynamically, according to the genetic diversity of each generation.

2.4. Bayesian Sampling Optimization (BO)

Bayesian Sampling Optimization, or Bayesian optimization for short, is a powerful technique for optimizing complex and costly objective functions. At its core, it combines two essential components: (1) a probabilistic surrogate model, typically a Gaussian Process (GP), and (2) an acquisition function. This approach offers an intelligent and efficient way to explore and exploit the search space, making it particularly valuable in scenarios where evaluating the objective function is time-consuming or resource-intensive, as in the case of model updating [22].

The probabilistic surrogate model (1) serves as a statistical representation of the underlying objective function. It leverages the principles of Bayesian inference to capture the

uncertainty associated with the function's behavior. The rationale for utilizing a statistical predictive model as a surrogate for the objective is to leverage its statistical characteristics for intelligent guidance in the task of sampling the objective function (as elaborated in more detail shortly afterwards). This task is easily achieved because drawing numerous samples from the surrogate model is cost-effective when compared to the expensive nature of evaluating the objective function itself. GPs are commonly employed as surrogate models due to their flexibility and ability to model complex non-linear relationships [23]. Indeed, these are by far among the most popular stochastic models used in the framework of BO and are as such employed in the current implementation as well. Nonetheless, several stochastic predictive models have been successfully employed as surrogates of the objective, such as random forests [24], deep neural networks [25], Bayesian neural networks [26], or ensembles of additive Gaussian Process models [27]. The GP is trained using observed data points (inputs and corresponding function values). These data points collectively constitute the surrogate model's training set. The GP's predictive distribution provides not only a point estimate of the objective function's value at any given input but also a measure of uncertainty or variance associated with the estimate.

The acquisition function (2) plays a pivotal role in guiding the optimization process [16]. It acts as a decision-making tool to select the next point to evaluate in the search space. In short, the acquisition function balances exploration and exploitation needs. It identifies promising areas to sample, emphasizing regions where samples drawn from the surrogate model exhibit high uncertainty (exploration) and/or where significant improvements in the objective function are likely (exploitation).

In many real-world optimization problems, the relationship between input dimensions is not uniform. The implemented strategy makes use of ARD kernels to address this issue by allowing the use of separate length scales (hyperparameters) for each dimension of the input space [28]. Each dimension is associated with a length scale, which represents how far apart input points in that dimension need to be for the output to become uncorrelated. This flexibility is particularly beneficial when dealing with anisotropic problems, where the scales of variation in different directions are dissimilar. The optimizer can therefore adapt more effectively to the specific characteristics of the problem, focussing its efforts on dimensions that have a more substantial influence.

Four different acquisition functions will be used and compared. These are briefly described hereinafter, but the reader is referred to the aforementioned literature. To this extent, the following notation is used:

$$\mathcal{D}_{1:t} = \{\mathbf{x}_{1:t}, f(\mathbf{x}_{1:t})\} \tag{1}$$

where $\mathcal{D}_{1:t}$ is the observations set, or sample, made of t observations in total and $\mathbf{x}_i$ is the vector of input parameters corresponding to the i-th observation. The length of $\mathbf{x}_i$ equals d, the number of dimensions of the updating problem, i.e., the number of updating parameters. Finally, $f(\mathbf{x}_{1:t})$ represents the values of the objective function sampled at $\mathbf{x}_{1:t}$.

The algorithm proceeds to select the next sampling point $\mathbf{x}_{t+1}$ by maximizing the acquisition function $a(\mathbf{x})$ (note that the actual objective value at $\mathbf{x}_{t+1}$, $f(\mathbf{x}_{t+1})$ is not guaranteed to be higher than the incumbent value $f(\mathbf{x}^+)$, where $\mathbf{x}^+ = \text{argmax}_{\mathbf{x}_i \in \mathbf{x}_{1:t}} f(\mathbf{x}_i)$):

$$\mathbf{x}_{t+1} = argmax_{\mathbf{x}}\, a(\mathbf{x}|\mathcal{D}_{1:t}) \tag{2}$$

For this task, $a(\mathbf{x})$ takes as input the predicted objective $\mu(\mathbf{x})$ and its uncertainty $\sigma(\mathbf{x})$ sampled from the Gaussian Process model, which serves as surrogate. Since the GP is computationally effective, this task requires limited (but non-negligible) computational effort.

The definition of the four acquisition functions that are to be compared with the simulated dataset is provided below.

Probability of Improvement (PI) [29] is defined as

$$
\begin{aligned}
\mathrm{PI}(\mathbf{x}) &= P(f(\mathbf{x}) \geq f(\mathbf{x}^+) + \xi) \\
&= \Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right)
\end{aligned}
\tag{3}
$$

where $\Phi(\cdot)$ is the CDF of the standard normal distribution.

This approach aims to quickly maximize the objective but tends to be overly exploitative due to its preference for low uncertainty predictions. This can lead to fast convergence rates but carries a high risk of running into local optima. Thus, a trade-off parameter ξ is introduced to balance exploration and exploitation. However, determining ξ to achieve adequate levels of exploration and convergence remains a challenging aspect [22].

Expected Improvement (EI) is defined as [22]

$$
\mathrm{EI}(\mathbf{x}) = \begin{cases} \sigma(\mathbf{x})\left[\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})}\Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})}\right) + \phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})}\right)\right] & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases}
\tag{4}
$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the CDF and the PDF of the of the standard normal distribution, respectively. This concept aims to identify the point with the highest expected improvement over the incumbent $f(\mathbf{x}^+)$, taking automatically into account both the improvement $\mu(\mathbf{x}) - f(\mathbf{x}^+)$ and the related uncertainty $\sigma(\mathbf{x})$ given by the predictive model.

Upper Confidence Bound (UCB), also known as Lower Confidence Bound (LCB) where minimization rather than maximization is concerned, is based on a very simple yet very effective concept. Presented by [30] in the "Sequential Design for Optimization" algorithm, this acquisition function consists in considering an upper (lower) confidence bound at **x**. The UCB function is defined as

$$
\mathrm{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x})
\tag{5}
$$

where κ is typically a positive integer number, which controls the bound width identified by the standard deviation $\sigma(\mathbf{x})$ and therefore the propensity to explore the optimization space. κ is taken as equal to 2, so that the confidence bound is about 95%. Also in this case, this approach leads to an automatic tradeoff between exploitation and exploration desires.

Finally, the modified version of Expected Improvement, (EI+), consists in introducing an overexploitation check at each iteration of the algorithm, as proposed by [31]. The goal is to recognize when the optimization procedure is overexploiting a certain area of the optimization space and force the algorithm to sample the objective away from this region. The following overexploitation check is performed on the selected point $\mathbf{x}_{t+1}$:

$$
\sigma(\mathbf{x}_{t+1}) < t_\sigma \sigma_A ,
\tag{6}
$$

where t_σ is an arbitrary coefficient that controls the overexploitation threshold, and σ_A is a small quantity of Gaussian noise added to the observations.

If the above condition is true, the kernel is modified by multiplying the GP's hyperparameters by the number of iterations. This leads to a higher variance in the space between observations, leading to higher acquisition values. After forcefully raising the variance, a new point x_{t+1} is selected for sampling and the check is performed once more. If the overexploiting condition is met again, the hyperparameters are multiplied by an additional factor of 10. This process continues until the overexploiting condition is no longer met or to a maximum of 5 times.

The implementation followed here is mostly based on the MatLab function bayesopt (https://it.mathworks.com/help/stats/bayesian-optimization-algorithm.html. Lat visited on 15 October 2023), which in turn derives from the research of Jones et al. [22] (where it is presented under the name Efficient Global Optimization—EGO), Gelbart et al. [32], and Snoek et al. [33].

In short, the BO procedure, in the current implementation, can be synthetized step by step as follows:

I. The seed points are randomly chosen within the optimization space, defined by the optimization bounds of each input parameter.

II. The optimization procedure is initialized by computing the objective function at these seed points.

III. The fitting of a Gaussian Process (GP) occurs by maximizing the marginal log-likelihood, which enables one to select the optimal set of hyperparameters $\boldsymbol{\theta}^{+}$. Moreover, a small amount of Gaussian noise σ^2 is added to the observations (such that the prior distribution has covariance $K(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) + \sigma^2 \boldsymbol{I}$).

IV. To maximize the acquisition function, several thousands of predictions $\mu_t(\mathbf{x}_*)$ are computed at the points $\mathbf{x}_*$, randomly chosen within the optimization space. Then, a selection of the best points is further improved with local search (to this end, the MatLab function fmincon (https://it.mathworks.com/help/optim/ug/fmincon.html. Last visited on 15 October 2023) is used in this application), among which the best point is finally chosen.

V. The objective function is computed at the point corresponding to the acquisition function maximum.

Hence, as described in point III, a newly fitted GP is used at each algorithm iteration. This represents one of the most computationally expensive (and thus time-consuming) aspects of the whole procedure. Nevertheless, as shown in Section 4, the gains due to a superior sampling strategy exceed the losses caused by the relatively cumbersome GP fitting. This, as will be demonstrated, results in an overall faster and more efficient solution than the other options inspected here.

Regarding further specific settings of the BO algorithm:

1. An ARD Matérn 5/2 kernel function is selected.

2. The input variables are log-transformed.

3. Four acquisition functions, as previously described, are considered. The best overall results, as will be shown, are achieved with the UCB acquisition function. All the options will be discussed in a dedicated paragraph.

Regarding the initialization settings, according to the established scientific literature:

1. The seed size was set to 50 points, following the advice of ref. [22] to set it as (at least) 10 *d*, where *d* is the number of dimensions of the optimization problem (i.e., updating parameters).

As a summary, Figure 1 provides flowcharts describing the basic workflow of the optimization techniques discussed so far.

2.5. Objective Function

Modal data (i.e., natural frequencies and mode shapes) are the most common features of FEMU. The minimization is performed between the model results and the target values until convergence. These target values correspond to the identified parameters in the case of an experimental dataset.

Concerning the mode shapes, the MAC—Modal Assurance Criterion [34]—value is considered to measure the coherence between the computed mode shapes and the target mode shapes. The natural frequencies and the MAC values used for updating are arranged together to compute the misfit between the computed response and the target response. To this extent, the following objective function (also known as *cost* or *penalty* function) is considered:

$$P = \sum_{i=1}^{N} \left| \frac{\omega_i^{targ/id} - \omega_i^{calc}}{\omega_i^{targ/id}} \right| + \sum_{i=1}^{N} \left(1 - \text{diag}\left(\text{MAC}\left(\boldsymbol{\phi}_i^{calc}, \boldsymbol{\phi}_i^{targ/id} \right) \right) \right), \tag{7}$$

where $\omega_i^{targ/id}$ and ω_i^{calc} are the *i*-th target (or identified) natural angular frequency and the *i*-th computed natural angular frequency, respectively, of the *N* modes used for updating, and $MAC\left(\phi_i^{calc}, \phi_i^{targ/id}\right)$ is the MAC value relative to the *i*-th computed mode shape ϕ_i^{calc} and the *i*-th target (or identified) mode shape $\phi_i^{targ/id}$. The first term of the cost function ensures that the natural frequencies calculated by the model are as close to the target ones as possible, while the second term ensures that the target mode shapes and the computed ones are correlated. MAC values close to 0 suggest little or no correlation between modes, while MAC values close to 1 indicate a good correlation. As such, if the computed modal data and target modal data are identical, the (always positive) objective function *P* is perfectly minimized at 0, which constitutes the global optimum of the function, providing the well-posedness of the updating problem. While this is indeed the case when updating against numerically simulated data, when using identified modal properties (i.e., experimental data), the global minimum of the cost function in the output space may lie far from zero. In fact, FE model deficiencies coupled with identified modal parameter inaccuracies lead to some ineluctable misfits between computed and measured modal responses.
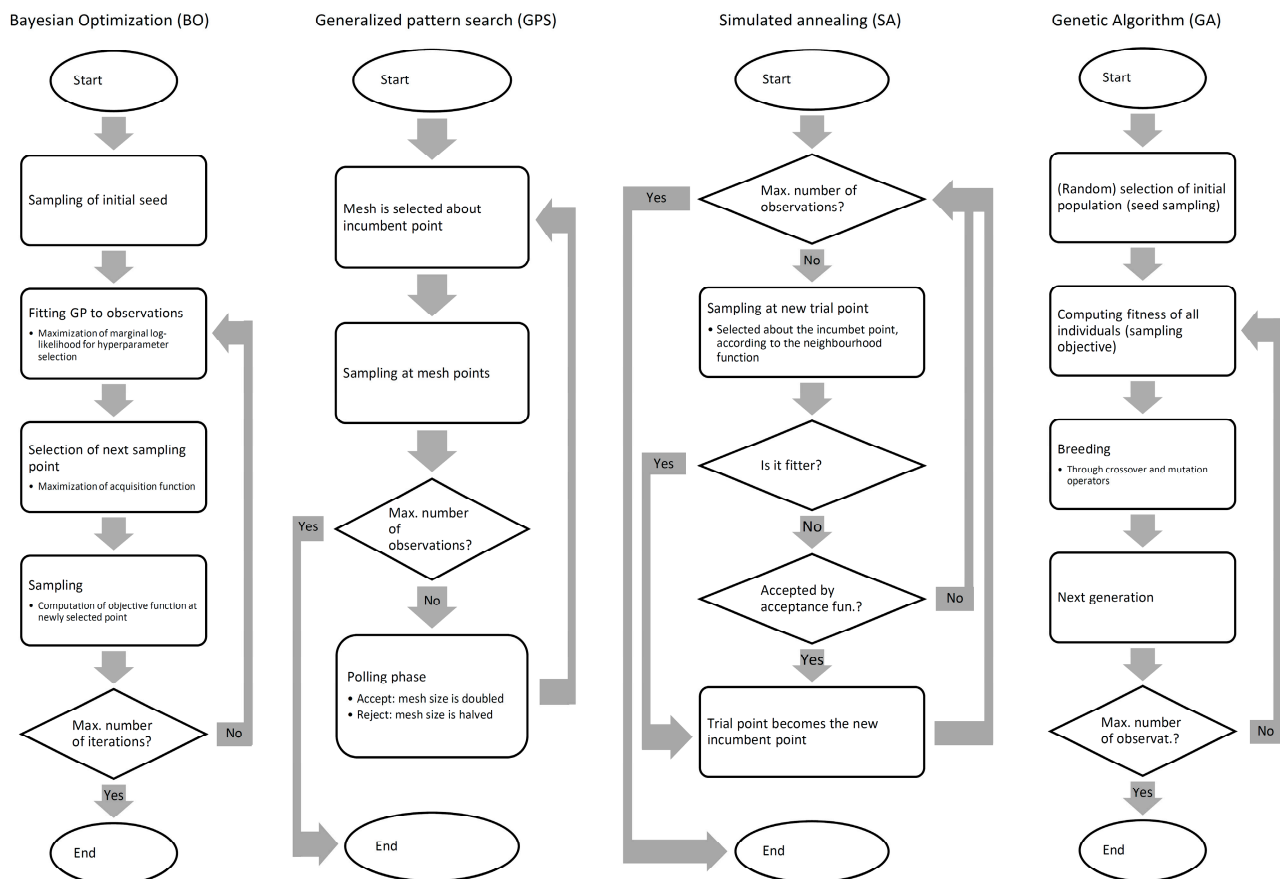

Figure 1. Flowcharts that describe the basic workflow of each optimization technique considered in this benchmark study.

2.6. Performance Metrics

Due to the different nature of the optimization techniques employed, some appropriate practices must be applied to promote a fair comparison of the algorithms:

1. Since SA and GA are stochastic/metaheuristic optimization techniques, the final optimization results are variable and non-deterministic. Hence, several optimization runs are performed with both algorithms: in the following case studies, the results shown stem from 10 different runs. Either the average or a representative case of the 10 executions is then considered.

2. The GPS algorithm must be initialized from a starting point. Obviously, the distance from the global optimum to the selected starting point greatly impacts the algorithm's efficiency and effectiveness. Hence, at each run, the algorithm is initialized at a different, randomly chosen point.

The four optimization techniques are compared by focussing on the accuracy level achieved according to the number of function evaluations performed by the algorithms. The achieved value of the objective function had been used as a measure of accuracy through the optimization process.

As fundamentally different optimization methods are being compared, the allowed number of function evaluations differs from case to case: in all case studies, GPS, SA, and GA are allowed to sample the objective function two times more than BO, as these techniques typically require a much greater sampling volume to achieve sufficient levels of accuracy. Admittedly, SA and GA, in particular, are optimization techniques that require a high number of iterations to perform at their best; hence, allowing more function evaluations makes the final results more comparable. On the other hand, this aspect is accounted for in the computation of the elapsed time.

Regarding Bayesian optimization, the number of iterations can be kept small to reduce computational time due to modelling and point selection tasks, without interfering negatively with the updating process. In fact, as will be shown in the results, BO outperforms these other candidates even with this arbitrary constraint of fewer computations allowed.

The relative error between target/identified and computed natural frequencies as well as the MAC values are reported for each case study. Similarly, for numerical case studies only, the relative error between the (known) target and the estimated parameters will be reported to assess the accuracy level in the input space of all optimization techniques.

Moreover, the root mean square relative error (RMSRE) is computed to help measure the overall response misfit of each algorithm. By considering all errors, this value (computed for both input space and output space quantities) gives a general measure of the level of accuracy reached by each technique. For results in the output space, the RMSRE is computed as

$$\text{RMSRE}_{\text{out}} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} f_{\text{rel},i}^2 + \frac{1}{n} \cdot \sum_{i=1}^{n} (1 - MAC_i)^2}, \qquad (8)$$

where $f_{\text{rel},i}$ is the relative error between the i-th natural frequency and its target value (so that $f_{rel,i} = \frac{f_i^{TAR} - f_i^{UPD}}{f_i^{TAR}}$), MAC_i is the MAC value of the i-th mode shape, and n is the number of modes considered for updating. For input space results, the RMSRE is simply given by

$$\text{RMSRE} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} X_{\text{rel},i}^2}, \qquad (9)$$

where $X_{\text{rel},i}$ is the relative error between the i-th updating parameter and its target value, and n is the number of updating parameters considered.

Finally, the total computational time of each optimization algorithm is used as a comparison metric. This allows one to prove the time-saving advantage of using a very efficient optimization technique, such as BO.

3. Case Study

The comparative study was performed on an aluminum frame structure, built in the research laboratory of Cranfield University as an Optical Test System (OTS) and dynamically tested indoors under controlled conditions. Independently from its specific destination of use, it shares many structural similarities with other metallic frame structures such as bridges [35].

Both experimental and numerically simulated data were employed. The latter case was intended for assessing the algorithm capabilities in a more controlled fashion. The experimental validation, on the other hand, proved the feasibility of the investigated FEMU approaches in the framework of real data acquisitions.

3.1. Cranfield Optical Test System

This case study originates from a unique OTS setup designed and assembled at the Precision Engineering Institute Laboratory of Cranfield University. This is discussed in detail in [36]; only the main aspects are recalled here for completeness.

This setup aimed to verify the form accuracy of meter-scale concave surfaces with nanometric precision. To satisfy this requirement, due to the radius of curvature of the test piece (an optical instrument), it was necessary to keep a 3.5 m distance between it and the measuring ZYGO DynaFiz™ laser interferometer. Hence, this structure (Figure 1) is of relevant engineering interest as it hosts a relatively heavyweight, very sensitive piece of equipment (with a ∼500 μm spatial resolution) at a relatively tall height. This instrument must be isolated from ambient vibrations (transmitted from the ground up) as much as possible; any potential excitement due to dynamic loads with frequencies close to the natural modes of the OTS would deteriorate its measurement capability.

The OTS structure is 4.25 m tall in total (accounting for the remaining part of the columns above the upper platen; see Figure 2b) and it is made of aluminum extruded profiles (see Figure 3c) bolted together (as shown in Figure 3a,b). The structure includes two structural platens, two dedicated (bespoke) precision positioning supports, five insulating and levelling feet (made of polyamide and bolted through the lower horizontal aluminum profiles), and a fixed support structure for the high-performance laser interferometer. Each main structural element was weighted before assembly; the details are reported in Table 1. For this reason, the density of the material was considered as a known value and not calibrated in the FEMU procedure. The test piece (420 × 420 × 40 mm) was made of ultra-low expansion (ULE) glass.
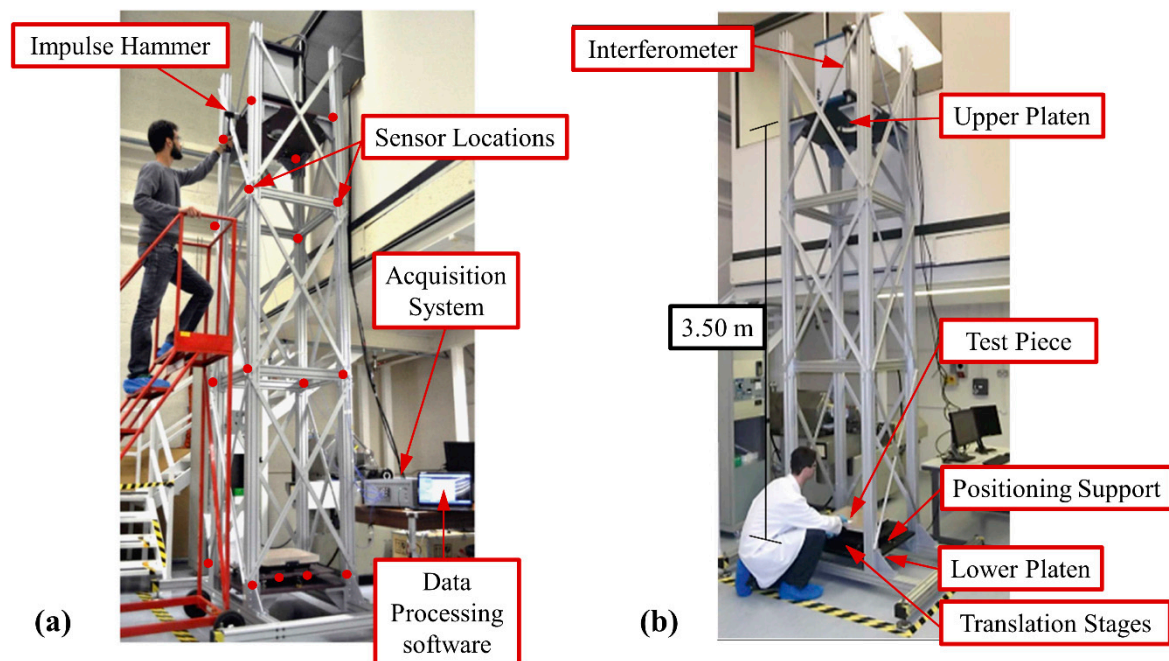

Figure 2. The OTS case study during experimental tests: (**a**) acquisition setup for the experimental modal analysis; (**b**) structural components.
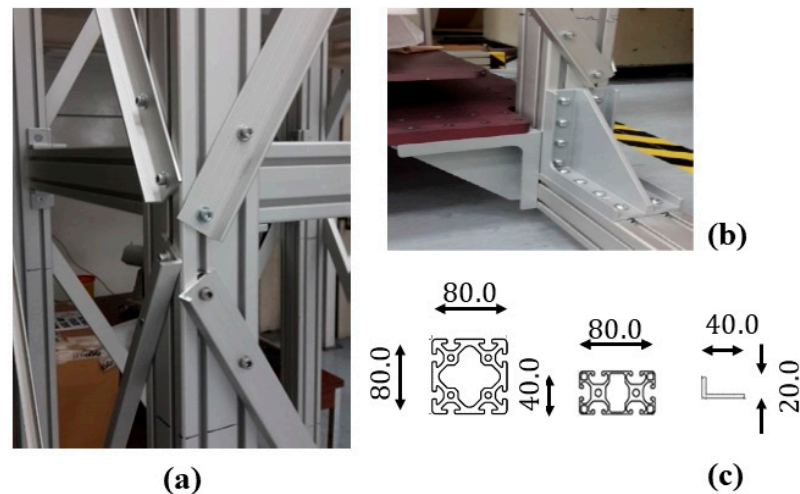
Figure 3. Details of the structure: (**a**) bolted joints between rods; (**b**) bolted connection between the base, one vertical pillar and the lower platen; (**c**) cross-section schematics of the aluminum profiles.

Table 1. Measured masses of the structural components.

Structural Component	Mass (kg)
Upper and lower aluminum platens	32 (each platen)
Interferometer and interferometer support structure	70
Test piece [1]	15.6
Test piece support structure [1]	16.6

[1] Summed up as the total mass of the lower platen (32.2 kg).

3.1.1. Acquisition Setup and Experimental Dataset

Both Experimental (input–output) and Operational (output-only) Modal Analyses (EMA/OMA) were performed. Only the first set of results is discussed here for brevity. For EMA, an impulse force test hammer (PBC Piezotronics model 086D20 ICP®) was performed as shown in Figure 2a, impacting the middle of the upper platen in the *x*- and *y*-axis directions. A super-soft tip was applied to excite frequencies in the range between 0 and 500 Hz. A set of four triaxial accelerometers (two 356A24 and two 356A25 ICP® PCB Piezotronics models) was applied for this investigation, with an LMS SCADAS III multichannel dynamic data acquisition system and the LMS test lab system identification software (https://plm.sw.siemens.com/en-US/simcenter/physical-testing/testlab/. Last visited on 15 October 2023). The sampling frequency was set to 360 Hz for all sensors. All recordings were then detrended and bandpass filtered. The complete testing equipment is portrayed in Figure 4.
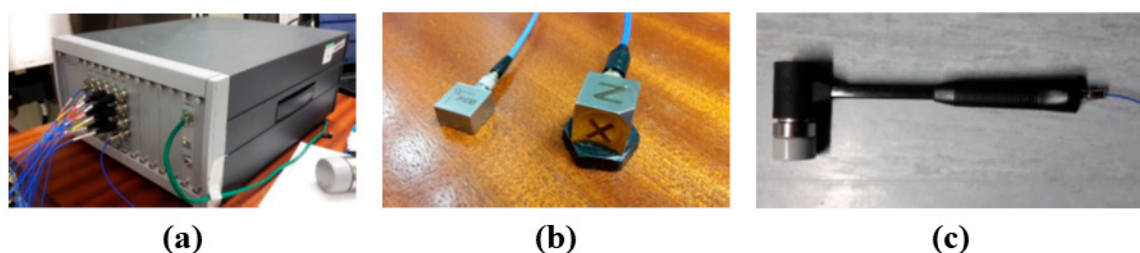

Figure 4. The Experimental Modal Testing equipment: (**a**) the 16-channel dynamic data acquisition (LMS SCADAS III); (**b**) Model 356A25 25 mV/g 1–5000 Hz and Model 356A24 10 mV/g 1–9000 Hz accelerometers; (**c**) impulse force test hammer (PCB Piezotronics model 086D20 ICP®) with interchangeable rubber tip.

The identified natural frequencies (up to the fourth mode, defined in the 0–50 Hz range) are reported in Table 2. The locations of the four sensors were changed between hammer hits as explained in [36], only maintaining one at a fixed location for reference and applying five hits per sensor layout. In total, 24 locations (highlighted by the red dots in Figure 2a) were used for the triaxial sensor, considering all the main frame joints and four points below the lower platen. Averaging the results, this returned experimental mode shape was defined over 72 output channels.

Table 2. Experimental dataset: identified natural frequencies (EMA).

Mode Number	Mode Description	f_n (Hz)
1	First bending mode along the *x*-axis	4.26
2	First bending mode along the *y*-axis	6.18
3	First axial (vertical) mode	22.52
4	First torsional mode	28.50

3.1.2. Finite Element Model

The FE Model utilized for this study is the same as that described in [36]. This was realized in ANSYS Mechanical APDL (Figure 5a). BEAM188 elements were utilized for the aluminum profiles and the interferometer support structure. SHELL181 elements were used for the upper and lower aluminum platens. All the bolted connections were assumed to be rigid joints. A lumped mass (MASS21) was used to simulate the interferometer, positioning it in its expected center of gravity and joining it to its support structure via six rigid links. COMBIN14 linear springs were applied at the five feet of the structure in the three directions (see Figure 5b). In the end, the complete model was composed of 2873 elements, totaling 2918 nodes.
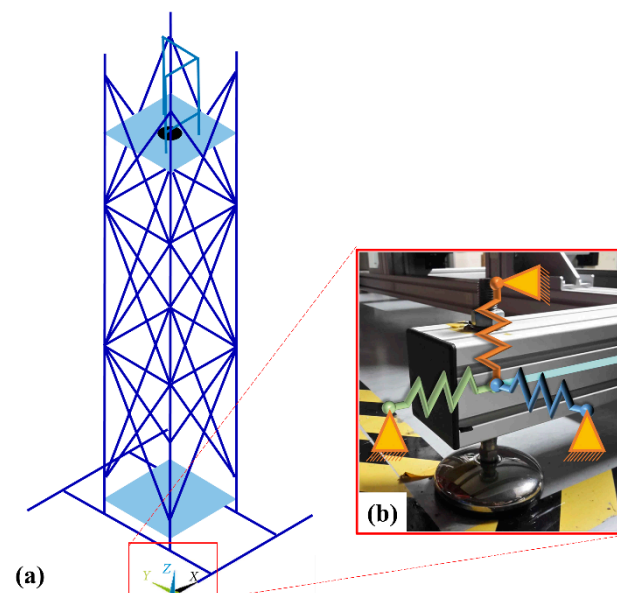

Figure 5. Finite Element Model of the OTS structure: (**a**) global overview; (**b**) detail of one levelling foot with its schematization.

The modal analysis of the structure is executed by employing the shifted-block Lanczos eigenvalue extraction method [37], a variation of the classical Lanczos algorithm. The number of modes to be extracted is set to 6, to keep the solution as computationally efficient as possible. The first four mode shapes of the structure are represented in Figure 6. These configurations exhibit topological congruence with the identified ones.
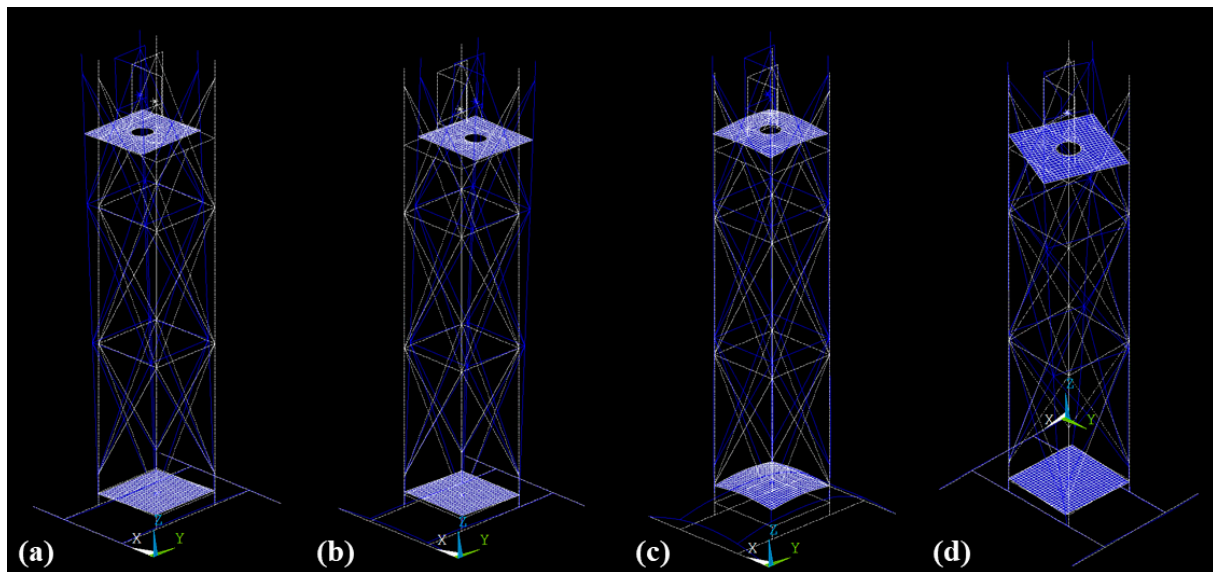
Figure 6. Mode shapes of the uncalibrated model: first bending mode along the *x*-axis (**a**); second bending mode along the *y*-axis (**b**); first axial (vertical) mode along the *z*-axis (**c**); and first torsional mode (**d**).

3.1.3. Model Updating Setup and Numerical Dataset

The following five parameters were considered for updating (see Table 3):

1. E_{Al}: the Young's modulus of the aluminum (assumed identical for all the beam and shell elements).
2. ν_{Al}: the Poisson's ratio of the same.
3. k_x, k_y, k_z: the linear stiffness of the springs at the feet of the structure, along the *x*-, *y*-, and *z*-axis, respectively (assumed identical for all feet).

Table 3. Parameters selected for updating and associated optimization bounds.

Parameters to Be Updated	Search Bounds		Measurement Unit
	lower bound	upper bound	
E_{Al}	30.00	150.00	GPa
ν_{Al}	0.3	0.4	-
k_x	1.20×10^3	3.00×10^4	Nm^{-1}
k_y	1.20×10^3	3.00×10^4	Nm^{-1}
k_z	1.20×10^3	3.00×10^4	Nm^{-1}

In the numerical simulation, these parameters are known in advance and thus named target parameters hereinafter. These quantities are in this case fixed a priori and chosen according to physically plausible values.

As mentioned earlier, the mass of the system is known and thus considered as a given ground truth. By varying stiffness-related quantities only, the risk of incurring ill-posedness issues of the updating problem is strongly reduced, enabling the assessment of the capabilities of the optimization techniques in a more controlled setting.

Also, this represents a quite realistic occurrence, as even for large and complex structures, the density of several building materials is the easiest property to estimate from material samples.

The optimization space is delimited by a lower bound and an upper bound, defined for each updating variable. The optimization bounds are tighter for the aluminum's elasticity modulus E_{Al}, as it is not expected to significantly deviate from the expected value of

70 GPa (retrieved from the material data sheet). Similarly, the optimization range of ν_{Al} is relatively limited as well. Instead, the optimization range for the three springs is wider, to account for higher uncertainty about the rigidity of the supports. Information given by the system-output data used for updating (first four natural frequencies and mode shapes) was found to be sufficient to guarantee uniqueness of the solution, as all optimization runs returned consistent results.

As mentioned earlier, to operate in a completely controlled fashion, not only the experimental dataset (described earlier) but also a numerically generated one was employed. In this latter case, the natural frequencies and mode shapes were obtained by running a modal analysis on the FE model set with the arbitrarily chosen values reported in Table 4. These will be referred to hereinafter as target input parameters and were intended to be similar, yet not identical (to avoid redundancy in the results), to the actual values of the calibrated model. Table 5 shows the target natural frequencies generated by the target input parameters for the numerical dataset.

Table 4. Values selected for the target input parameters.

Target Input Parameters	Target Value
E_{Al}	70.00 GPa
ν_{Al}	0.32
k_x	1.00×10^4 Nm^{-1}
k_y	2.00×10^3 Nm^{-1}
k_z	2.00×10^4 Nm^{-1}

Table 5. Numerical dataset: target natural frequencies (numerically generated).

Mode Number	Mode Description	f_n (Hz)
1	First bending mode along the x-axis	4.39
2	First bending mode along the y-axis	6.50
3	First axial (vertical) mode	22.70
4	First torsional mode	29.15

4. Results

The case study reported here serves well the aim of showcasing the potential of the several algorithms, given the nature of the optimization problem. On the one hand, the rather simple geometry and building material of the system allow one to reduce the sources of uncertainty. On the other hand, as very heterogeneous parameters are being optimized, namely material properties and spring stiffnesses, the problem is expected to show different sensitivity to each updating parameter. Hence, the algorithms must be able to deal with different search ranges.

The use of numerically simulated and experimental data plays an important role in testing and assessing the algorithms in analysis. The numerical simulation provides a controlled environment, allowing for systematic testing of the algorithms under precisely defined conditions. This controlled setting enables to evaluate the algorithmic performance and validate the effectiveness of each optimization strategy in the idealized scenario, where the actual (target) values of the updating parameters are known. On the other hand, the experimental data setup introduces the complexities and uncertainties inherent of real-world situations, reflecting the practical challenges of dealing with errors and noise due to both model deficiencies and modal identification uncertainties. This provides insights into the robustness, adaptability, and accuracy of the four benchmarked methods when confronted with the natural imperfections of physical measurements and modelled response of the system.

4.1. Results for the Numerically Simulated Data

As mentioned, the comparison between BO, GPS, GA, and SA is conducted through a comparison of the results in the output and the input space. At this stage, no noise is added to the target modal parameters (i.e., natural frequencies and mode shapes), to promote fair comparison between the four methods in terms of performance of accuracy. This idealized scenario enables to isolate the optimization problem from other well-known sources of error in model updating problems. Conversely, these will be naturally present in the benchmarks conducted with experimental data. For BO, all results shown in this and the following section refer to the Upper Confidence Bound option as specified in Section 2.4. A more detailed discussion will be reported later in a dedicated paragraph.

Results relative to the output, in terms of (best) cost function values and the estimated modal data, are shown in Table 6. The table reports the relative errors between the updated natural frequencies and the known target values (generated by the set of target parameters), the MAC values, and the root mean square relative error (RMSRE), which, by considering both the modal parameters, represents a sort of "final score" that helps to measure the overall response misfit of each algorithm. The relative errors are expressed in percentage and computed, for each n-th mode, as $(f_n^{UPD} - f_n^{TAR})/f_n^{TAR}$, where f_n^{UPD} is the value estimated by the FE model at the end of the update and f_n^{TAR} is the corresponding target.

Table 6. Optimization results in the output space, numerical dataset.

Mode Number	f_n^{TAR} (Hz)	GPS (200 Fun. Eval.)			SA (200 Fun. Eval.)			GA (200 Fun. Eval.)			BO (100 Fun. Eval.)		
		f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)	f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)	f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)	f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)
1	4.39	4.38	−0.228	1.0000	4.41	0.456	1.0000	4.40	0.228	1.0000	4.39	0.000	1.0000
2	6.50	6.53	0.462	0.9999	6.48	−0.308	1.0000	6.54	0.615	1.0000	6.51	0.154	1.0000
3	22.70	22.66	−0.176	0.9995	22.54	−0.705	1.0000	22.74	0.176	0.9998	22.74	0.176	1.0000
4	29.15	29.84	2.367	0.9998	29.89	2.539	0.9999	29.52	1.269	1.0000	29.17	0.069	1.0000
RMSRE (%)		2.16			1.43			1.92			0.29		
Best cost fun.		0.0612			0.0502			0.0511			0.0070		

Upon evaluating the attained cost function optima and the RMSRE concerning the modal data, it becomes apparent that all algorithms demonstrate satisfactory, albeit not exceptional, minimization performance. The least favorable outcome is observed when employing GPS. However, even in this least favorable scenario, the objective value does not deviate significantly from zero (i.e., the known global minimum).

SA and GA perform similarly, yielding acceptable values of RMSRE, suggesting that these algorithms might have successfully identified the global optimum. Concerning these two optimization techniques, all MAC values are found to be very close to 1, suggesting a very high correlation between the computed mode shapes and the target ones. Regarding the natural frequencies, SA reveals a relatively slightly higher level of error for f_4 while still showing very good agreement with the remaining target frequencies.

Bayesian optimization (BO) truly distinguishes itself, presenting superior results in terms of the best cost function value achieved and, consequently, the most precise alignment between the target and updated modal data. Indeed, the accuracy reached for both mode shapes and natural frequencies (which, once again, take up most of the discrepancy between target and computed data) is quite remarkable, especially considering that these results were obtained with only 100 iterations (50% less than the other algorithms).

The results relative to the input space, i.e., in terms of the updated parameters, are displayed in Table 7. The table reports, for each updating parameter, the target value, as this is known (having been manually set). The relative error is computed between them and the estimates obtained from the FE model when updated by using the four optimization techniques. As before, the formulation for the error is as $(P_i^{UPD} - P_i^{TAR})/P_i^{TAR}$, for the

i-th parameter P. In general, the parameters' estimations show a much higher error when compared to the results obtained in the output space (i.e., updated modal data), denoting that the objective function of this specific updating setup is scarcely sensitive to even significant changes of some parameters. That reflects a generally low sensitivity of the model to these parameters, which is overall not uncommon for FEMU [2].

Table 7. Optimization results in the input space, numerical dataset.

Parameter P	Target Value P^{TAR}	GPS (200 Fun. Eval.)		SA (200 Fun. Eval.)		GA (200 Fun. Eval.)		BO (100 Fun. Eval.)	
		Updated Value P^{UPD}	Error (%)	Updated Value P^{UPD}	Error (%)	Updated Value P^{UPD}	Error (%)	Updated Value P^{UPD}	Error (%)
E_{Al}	70.00 GPa	72.00	2.857	72.36	3.371	72.28	3.257	70.37	0.529
ν_{Al}	0.32	0.40	25.000	0.32	0.000	0.37	15.625	0.34	6.250
k_x	1.00×10^4 Nm^{-1}	3.00×10^4	200.000	2.47×10^4	147.000	8.11×10^3	-18.900	9.47×10^3	-5.300
k_y	2.00×10^3 Nm^{-1}	3.12×10^3	56.000	2.08×10^3	4.000	2.44×10^3	22.000	1.93×10^3	-3.500
k_z	2.00×10^4 Nm^{-1}	1.46×10^4	-27.000	1.31×10^4	-34.500	1.53×10^4	-23.500	1.98×10^4	-1.000
RMSRE (%)		94.33		67.68		18.08		3.62	

More specifically, the dynamic response of the structure is influenced in a very limited way by the springs that model the rigidity of the supports, i.e., by its boundary conditions, as the largest deviations are found in these parameters. On the other hand, the estimation of the material Young's modulus is quite accurate for all the algorithms. In particular, GPS, SA, and GA attain good estimations of E_{Al} but fail at achieving correct values of k_x, k_y, and k_z, thus generating high errors. BO is the only technique that attained accurate results even for these parameters, in agreement with the results obtained in the output space.

In fact, for all the reasons mentioned above, the values of RMSRE (computed according to Equation (3)) are sensibly higher than their counterparts in Table 6 (computed according to Equation (2)). They also show a much larger discrepancy between the four candidates. This returns a well-defined ranking, with BO followed by GA, SA, and GPS.

The total optimization time employed by each algorithm is reported in Table 8. For BO, the total time required to attain the solution of the secondary optimization problems (modelling and point selection time) was 73 s, thus almost 10% of the total elapsed time. Nevertheless, BO runs sensibly faster than GPS, SA, and GA, given the halved amount of function evaluations. The other three candidates, instead, seem to have almost the same computational needs for equal conditions. In particular, the total optimization time practically equals the time used to compute the cost function times the number of evaluations. All runs were performed on an Intel® Core™ i7- 12700H CPU with a 2.70 GHz base frequency.

Table 8. Elapsed optimization time, numerical dataset.

	GPS (200 Fun. Eval.)	SA (200 Fun. Eval.)	GA (200 Fun. Eval.)	BO (100 Fun. Eval.)
Total optimization time [s]	1358	1363	1378	756

As proved earlier in this subsection, BO returned consistently the best results, with half the sampling volume needed by the other algorithms and slightly more than half of the total elapsed time. Especially in the input domain (i.e., for the estimation of the input parameters), Bayesian optimization reached by far the best accuracy on all parameters, keeping the error levels very low. In summary, it significantly outperformed the other global optimization techniques considered. Also for this reason, a more detailed description of this technique's results is provided here.

4.1.1. Results of the Four BO Acquisition Functions

Figure 7 shows the results obtained by performing BO on the case study with the four acquisition functions considered in this study—PI, EI, EI+, and UCB. As mentioned earlier, UCB returns the best fitting in the end; however, a comparison with the other options gives important insights into the potential of BO.

In particular, it can be noticed how PI converges to the optimum much more rapidly than the other acquisition functions, which instead tend to further explore the optimization space.

This is also evident in Figure 8 (which reports the best objective value obtained during the optimization), where one can see the lower value of the first function evaluation performed by PI with respect to the other functions. Given the good model validation assessed in the previous step, PI returns very high accuracy within a very limited amount of function evaluations. On the other hand, while retaining a higher propensity to explore at early stages, Upper Confidence Bound (Lower confidence bound in the Figure) gradually increases the accuracy of the solution towards the final phases of the optimization process. Ultimately, this causes UCB to return the best result in terms of the accuracy of the final solution among the four options, even if by a small margin.
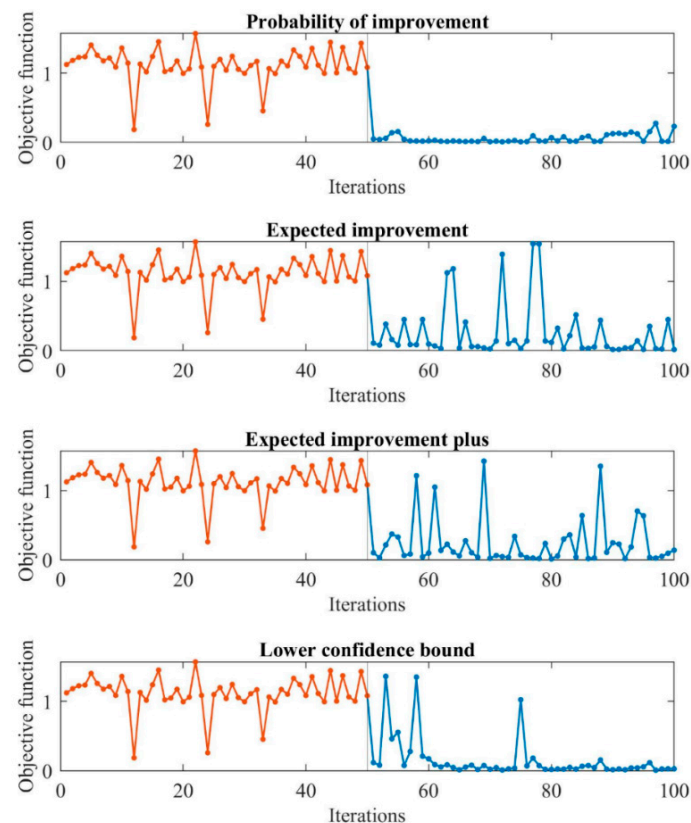

Figure 7. The behavior of the Bayesian optimization throughout the optimization process. The objective value at the randomly sampled seed points is displayed in red, while the objective at points selected by the acquisition function is displayed in blue.
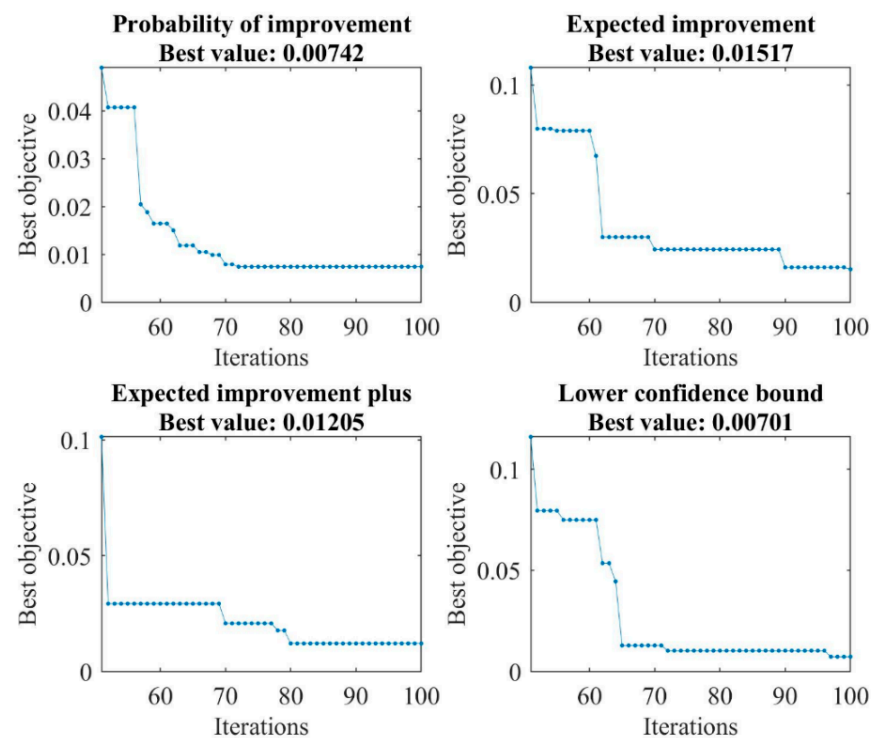
Figure 8. The best objective value achieved during the optimization process when using PI (**top left**), EI (**top right**), EI+ (**bottom left**), and Lower Confidence Bound (LCB) (**bottom right**).

4.1.2. Hyperparameters of Fitted Gaussian Process

Since automatic relevance determination (ARD) kernel functions are employed, it is possible to retrieve the length scale of each updating parameter from the hyperparameters of the Gaussian Process fitted to the initial seed. The length scales (reported in Table 9) provide information about the sensitivity of the problem to the updating variables. Note that a high length scale suggests low sensitivity, while a low length scale indicates high sensitivity. As expected, the five length scales do not have the same order of magnitude: the parameter which majorly affects the problem (i.e., the system modal response) is the aluminum elasticity modulus, as expected for obvious reasons. Indeed, the material stiffness usually has the most significant impact on the resulting modal properties. On the contrary, and quite reasonably, the problem is not very sensitive to changes in the Poisson's ratio. Concerning the three springs, k_y is found to have a greater impact on the output, while k_x and k_z feature lower sensitivity. The length scales, computed by maximizing the log-likelihood, are relative to the 50 points randomly sampled for the seed. Since the problem sensitivity to the updating parameters varies across the input space, slightly different length scales are obtained when changing the initial seed of points. Also, kernel hyperparameters adapt during the optimization process, as more sampling points are added to the objective observations.

Table 9. Length scales of the input parameters as obtained by maximizing the marginal log-likelihood in BO.

Parameter	Length Scale (-)
E_{Al} (GPa)	0.59
ν_{Al} (-)	6239.04
k_x $(\mathrm{Nm^{-1}})$	23.28
k_y $(\mathrm{Nm^{-1}})$	0.98
k_z $(\mathrm{Nm^{-1}})$	8.29

4.2. Results for the Experimental Data

In this subsequent set of results, the endeavor pertains to the recalibration of the FE model utilizing actual experimental data. In this specific instance, the veritable values of material parameters remain unknown, rendering the comparison to be executed through two distinctive lenses:

1. A comparison in absolute terms is performed, addressing the disparity in modal properties derived from the four calibrated FE models and those inferred from the empirical acquisitions.

2. In relative terms, the assessment is based on the contrast between the parameters estimated by the four candidate models.

Due to the utilization of real-world measurements, it is essential to acknowledge the presence of inherent uncertainties arising from modelling and system identification contributing to ineluctable error within the cost function. The cost function is therefore bound to persist with a non-zero value. Nevertheless, this setup allows for conducting a further comparison between Bayesian optimization and the optimization algorithms utilized as benchmark standards. In fact, this enables to assess how well the algorithms can automatically deal with experimental data that are naturally affected by measurement noise and errors.

The results for this comparative evaluation are presented in Table 10 for the input space and Table 11 for the output space, in the same fashion as the results presented for the synthetic data scenario.

Table 10. Optimization results in the output space, experimental dataset.

Mode Number	f_n^{TAR} (Hz)	GPS (200 Fun. Eval.)			SA (200 Fun. Eval.)			GA (200 Fun. Eval.)			BO (100 Fun. Eval.)		
		f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)	f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)	f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)	f_n^{UPD} (Hz)	Δf Error (%)	MAC (-)
1	4.26	4.26	−0.23	0.92	4.28	0.23	0.92	4.24	−0.7	0.92	4.26	−0.23	0.92
2	6.45	6.45	4.37	0.95	6.4	3.56	0.95	6.39	3.4	0.95	6.36	2.91	0.95
3	22.31	22.31	−0.93	0.95	22.1	−1.86	0.94	22.07	−1.99	0.95	22.1	−1.87	0.95
4	28.52	28.52	0.07	0.95	28.91	1.44	0.95	28.51	0.04	0.94	28.53	0.11	0.95
RMSRE (%)		4.46			4.59			4.55			4.34		
Best cost fun.		0.1205			0.1313			0.1229			0.1201		

Table 11. Optimization results in the input space, experimental dataset.

Parameter P	GPS (200 Fun. Eval.)	SA (200 Fun. Eval.)	GA (200 Fun. Eval.)	BO (100 Fun. Eval.)
E_{Al}	65139	66302	64310	65131
ν_{Al}	0.3311	0.3103	0.3068	0.30883
k_x	6928	24341	18102	28696
k_y	8976	6234	8020	3242.5
k_z	29456	18234	26478	22939

$E_{Al}\nu_{Al}k_x k_y k_z$ Upon an examination of the estimated input parameters, it is evident that the parameter exhibiting the highest degree of consistency is the elasticity modulus of aluminum. This parameter emerges as the most responsive within the chosen parameters for optimization. Conversely, parameters related to the axial spring stiffnesses exhibit a conspicuous variability among the four optimization techniques employed. Notably, there is a discernible trend wherein higher stiffness values are assigned to the springs in the x and z directions, while lower values are allocated to those in the y direction. This tendency can be attributed to the inherent geometry of the structural configuration, characterized by the limited bending and torsional stiffness in the lower support beams of the structure. The

Poisson's ratio, in a similar fashion, exhibits elevated variability, stemming from its status of least sensitive parameter in the context of the updating problem, as underscored by the GP hyperparameters derived from the numerically simulated dataset.

When considering the final values pertaining to natural frequencies and MAC values, all the employed algorithms yield modest and acceptable errors. It is worth emphasizing that the sensitivity of the objective function remains relatively low. This observation signifies that even minor variations in the selected modal parameters can engender substantial alterations in the input parameters of interest.

However, given the inherent limitations of the specific updating problem driven by empirical data, Bayesian Optimization distinguishes itself again, by achieving the most favorable alignment between identified modal parameters and their computed counterparts. Moreover, it accomplishes this while demonstrating notably higher computational efficiency, with a runtime of approximately half that of the alternative optimization strategies.

5. Conclusions

In this comparative analysis, four optimization algorithms—Generalized Pattern Search (GPS), Simulated Annealing (SA), Genetic Algorithm (GA), and Bayesian Sampling Optimization (BO)—were compared for Finite Element Model Updating (FEMU) purposes. A metallic truss structure, built in the facilities of Cranfield University, was used as a benchmark problem. Both numerical (with known ground truth) and experimental (with unknown parameters) datasets were used.

The proposed case study represents an interesting application, as both a detailed FE model and a large set of experimental data for its calibration were available. The structure can be seen as a well-representative benchmark problem since:

(i.)   Its geometry and material are very similar to many metallic truss structures, commonly found in civil engineering applications.
(ii.) It was dynamically tested under a strictly controlled environment and operating conditions.
(iii.) Its unknown parameters comprise both the structure's properties (E_{Al}, ν_{Al}) and its boundary conditions (k_x, k_y, k_z).

The calibration of these five unknowns represented a non-trivial task, as they spanned a five-dimensional search space, with search ranges of different magnitudes. In this regard, it is important to remember that BO returned not only the updated parameters of the FE model but also the length scale for each of them. These values can be directly used for sensitivity analysis, bypassing the need for a dedicated study.

In brief, Bayesian Sampling Optimization clearly surpassed the other options in terms of accuracy and computational efficiency. The superior accuracy is visible both in the output domain (more accurate numerical simulations of the natural frequencies and mode shapes, up to the structure's sixth mode, both for the numerical and the experimental dataset) and especially in the input space (more accurate estimates of the known input parameters for the numerical dataset).

The other three candidates performed quite similarly in terms of computational time. In the output domain, SA and GA performed better than GPS but, again, quite similarly to each other. Finally, the results in the input parameters' estimations allow one to outline a well-defined ranking, with Genetic Algorithm in second place, Simulated Annealing in third place, and Generalized Pattern Search in fourth (last) place. Nevertheless, the difference between Bayesian Sampling Optimization and the other options, including Genetic Algorithm, is very relevant. Thus, the use of BO is strongly recommended by the authors. Bayesian optimization presents fewer challenges in the selection of specific algorithm parameters compared to techniques such as SA or GA. However, the somewhat intricate workflow may impede its implementation for inexperienced users. Another technical aspect worth noting is the limited scalability of Gaussian Processes, as the computational cost is cubic with respect to the number of data points. This places a practical constraint on the maximum number of algorithm iterations when utilizing such models in the BO framework.

In fact, as the number of observations increases, the algorithm becomes progressively more computationally expensive.

The results shown in this research work derive from a single case study, for which both a detailed FE model and a large set of experimental data for its calibration were available. The structure can be seen as a well-representative benchmark problem since its unknown parameters comprise both the structure's properties and its boundary conditions. However, the findings of this research work can be applied to any structural typology, building material, or boundary condition, ranging from, e.g., other metallic truss structures to masonry or reinforced concrete buildings.

Author Contributions: Conceptualization, M.C. and L.Z.F.; methodology, M.C. and L.Z.F.; software, D.R., M.C. and L.Z.F.; validation, D.R., M.C. and L.Z.F.; formal analysis, D.R. and M.C.; resources, L.Z.F.; data curation, D.R., M.C. and L.Z.F.; writing—original draft preparation, D.R. and M.C.; writing—review and editing, L.Z.F.; visualization, D.R. and M.C.; supervision, M.C. and L.Z.F.; project administration, L.Z.F. All authors have read and agreed to the published version of the manuscript.

References

1. Haag, S.; Anderl, R. Digital Twin—Proof of Concept. *Manuf. Lett.* **2018**, *15*, 64–66. [CrossRef]
2. Boscato, G.; Russo, S.; Ceravolo, R.; Fragonara, L.Z. Global Sensitivity-Based Model Updating for Heritage Structures. *Comput.-Aided Civ. Infrastruct. Eng.* **2015**, *30*, 620–635. [CrossRef]
3. Friswell, M.; Mottershead, J.E. *Finite Element Model Updating in Structural Dynamics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1995.
4. Friswell, M.I. Damage Identification Using Inverse Methods. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2007**, *365*, 393–410. [CrossRef] [PubMed]
5. Mottershead, J.E.; Friswell, M.I. Model Updating In Structural Dynamics: A Survey. *J. Sound Vib.* **1993**, *167*, 347–375. [CrossRef]
6. Girardi, M.; Padovani, C.; Pellegrini, D.; Porcelli, M.; Robol, L. Finite Element Model Updating for Structural Applications. *J. Comput. Appl. Math.* **2020**, *370*, 112675. [CrossRef]
7. Ereiz, S.; Duvnjak, I.; Fernando Jiménez-Alonso, J. Review of Finite Element Model Updating Methods for Structural Applications. *Structures* **2022**, *41*, 684–723. [CrossRef]
8. Nicoletti, V.; Gara, F. Modelling Strategies for the Updating of Infilled RC Building FEMs Considering the Construction Phases. *Buildings* **2023**, *13*, 598. [CrossRef]
9. Arezzo, D.; Quarchioni, S.; Nicoletti, V.; Carbonari, S.; Gara, F.; Leonardo, C.; Leoni, G. SHM of Historical Buildings: The Case Study of Santa Maria in Via Church in Camerino (Italy). *Procedia Struct. Integr.* **2023**, *44*, 2098–2105. [CrossRef]
10. Xiao, F.; Zhu, W.; Meng, X.; Chen, G.S. Parameter Identification of Structures with Different Connections Using Static Responses. *Appl. Sci.* **2022**, *12*, 5896. [CrossRef]
11. Xiao, F.; Zhu, W.; Meng, X.; Chen, G.S. Parameter Identification of Frame Structures by Considering Shear Deformation. *Int. J. Distrib. Sens. Netw.* **2023**, *2023*, 6631716. [CrossRef]
12. Xiao, F.; Sun, H.; Mao, Y.; Chen, G.S. Damage Identification of Large-Scale Space Truss Structures Based on Stiffness Separation Method. *Structures* **2023**, *53*, 109–118. [CrossRef]
13. Torczon, V. On the Convergence of Pattern Search Algorithms. *SIAM J. Optim.* **2006**, *7*, 1–25. [CrossRef]
14. van Laarhoven, P.J.M.; Aarts, E.H.L. Simulated Annealing. In *Simulated Annealing: Theory and Applications*; Springer: Dordrecht The Netherlands, 1987; pp. 7–15. [CrossRef]
15. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
16. Brochu, E.; Cora, V.M.; De Freitas, N. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv* **2010**, arXiv:1012.2599.
17. Hooke, R.; Jeeves, T.A. "Direct Search" Solution of Numerical and Statistical Problems. *J. ACM (JACM)* **1961**, *8*, 212–229. [CrossRef]
18. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
19. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **2004**, *21*, 1087. [CrossRef]

20. Ingber, L. Adaptive Simulated Annealing (ASA): Lessons Learned. *Control Cybern.* **2000**, *25*, 32–54. [CrossRef]

21. Civera, M.; Pecorelli, M.L.; Ceravolo, R.; Surace, C.; Zanotti Fragonara, L. A Multi-objective Genetic Algorithm Strategy for Robust Optimal Sensor Placement. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 1185–1202. [CrossRef]

22. Jones, D.R.; Schonlau, M.; Welch, W.J. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13*, 455–492. [CrossRef]

23. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; Massachusetts Institute of Technology, Ed.; MIT Press: Cambridge, MA, USA, 2006; ISBN 026218253X.

24. Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Sequential Model-Based Optimization for General Algorithm Configuration. In Proceedings of the Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, 17–21 January 2011; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6683, pp. 507–523. [CrossRef]

25. Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Mostofa, M.; Patwary, A.; Adams, R.P. Scalable Bayesian Optimization Using Deep Neural Networks. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015.

26. Springenberg, J.T.; Klein, A.; Falkner, S.; Hutter, F. Bayesian Optimization with Robust Bayesian Neural Networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4134–4142.

27. Wang, Z.; Gehring, C.; Kohli, P.; Jegelka, S. Batched Large-Scale Bayesian Optimization in High-Dimensional Spaces. In Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS), Lanzarote, Spain, 9–11 April 2018.

28. Rasmussen, C.E.; De, H.M. Gaussian Processes for Machine Learning (GPML) Toolbox Hannes Nickisch. *J. Mach. Learn. Res.* **2010**, *11*, 3011–3015.

29. Kushner, H.J. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *J. Basic Eng.* **1964**, *86*, 97–106. [CrossRef]

30. Cox, D.D.; John, S. A Statistical Method for Global Optimization. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Chicago, IL, USA, 18–21 October 1992; pp. 1241–1246. [CrossRef]

31. Bull, A.D. Convergence Rates of Efficient Global Optimization Algorithms. *J. Mach. Learn. Res.* **2011**, *12*, 2879–2904.

32. Gelbart, M.A.; Snoek, J.; Adams, R.P. Bayesian Optimization with Unknown Constraints. *arXiv* **2014**, arXiv:1403.5607. [CrossRef]

33. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *Adv. Neural Inf. Process. Syst.* **2012**, *4*, 2951–2959. [CrossRef]

34. Allemang, R.J.; Brown, D.L. A Correlation Coefficient for Modal Vector Analysis. In Proceedings of the 1st International Modal Analysis Conference (IMAC 1982), Orlando, FL, USA, 8–10 November 1982; pp. 110–116.

35. Xiao, F.; Hulsey, J.L.; Chen, G.S.; Xiang, Y. Optimal Static Strain Sensor Placement for Truss Bridges. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147717707929. [CrossRef]

36. Golanó, P.G.; Zanotti Fragonara, L.; Morantz, P.; Jourdain, R. Numerical and Experimental Modal Analysis Applied to an Optical Test System Designed for the Form Measurements of Metre-Scale Optics. *Shock Vib.* **2018**, *2018*, 3435249. [CrossRef]

37. Grimes, R.G.; Lewis, J.G.; Simon, H.D. A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems. *Soc. Ind. Appl. Math.* **1994**, *15*, 228–272. [CrossRef]