Engineering Structures 195 (2019) 373–381

Contents lists available at ScienceDirect

Engineering Structures

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

Finite element model updating using deterministic optimisation: A global pattern search approach

Benedikt Hofmeister*, Marlene Bruns, Raimund Rolfes

Leibniz University Hannover/ForWind, Institute of Structural Analysis, Appelstraße 9A, 30167 Hannover, Germany

ARTICLE INFO

Keywords:
Deterministic optimisation
Derivative-free methods
Finite element method
Model updating
Wind turbine
Rotor blades

ABSTRACT

With this work, we present a novel derivative-free global optimisation approach for finite element model updating. The aim is to localise structural damage in a wind turbine rotor blade. For this purpose, we create a reference finite element model of the blade as well as a model with a fictitious damage. To validate the approach, we use a model updating scheme to locate the artificially induced damage. This scheme employs numerical optimisation using the parameterised finite element model and an objective function based on modal parameters.

Metaheuristic algorithms are the predominant class of optimisers for global optimisation problems. With this work, we show that deterministic approaches are competitive for engineering problems such as model updating. The proposed optimisation algorithm is deterministic and a generalisation of the pattern search algorithm. It picks up features known from local deterministic algorithms and transfers them to a global algorithm. We demonstrate the convergence, discuss the numerical performance of the proposed optimiser with respect to several analytical test problems and propose a possible trade-off between parallelisation and convergence rate. Additionally, we compare the numerical performance of the proposed deterministic algorithm concerning the model updating problem to the performance of well-established metaheuristic and local optimisation algorithms.

The introduced algorithm converges quickly on test functions as well as on the model updating problem. In some cases, the deterministic algorithm outperforms metaheuristic algorithms. We conclude that deterministic optimisation algorithms should receive more attention in the field of engineering optimisation.

1. Introduction

For optimisation tasks considering non-linear problems, derivative-free global algorithms are particularly suited. Objective functions of such problems often involve transient numerical simulations or discrete and non-linear evaluations. This is why it is usually not possible to find a direct solution for the derivative of such objective functions.

We concentrate on derivative-free algorithms, since obtaining derivatives in a numerically complex design variable space is challenging. Indeed, derivatives can easily be obtained numerically by using single-sided or symmetric sampling around a base point. The Hessian matrix needed for sequential quadratic programming [1] is commonly obtained by this method. However, numerical noise and the difficulty to receive an appropriate value for the step size necessitate some numerical experiments to yield a stable optimisation. Derivative-free methods are thus desirable due to the numerical robustness they provide.

Most commonly used derivative-free algorithms are metaheuristic. This means that they rely on pseudo-random numbers in order to stochastically explore the design variable space of the underlying problem. Examples of this class of algorithms are genetic algorithms [2], particle swarm optimisation [3] or harmony search [4]. More recent contributions also include algorithms inspired by biological phenomena and swarm intelligence like whale optimisation [5], bacterial foraging optimisation [6], anarchic society optimisation [7] or social-spider optimisation [8]. The random sampling employed in these algorithms ensures an evenly distributed evaluation of the objective function. On the contrary, a derivative-free deterministic optimisation algorithm needs a way to generate a sufficient distribution of sampling points by means of a non-stochastic expression.

Studies with a large variety of benchmark problems and optimisation algorithms have shown that deterministic algorithms can achieve a performance similar to metaheuristic approaches, in some cases even exceeding them [9]. On these benchmark problems, the deterministic DIRECT algorithm [10] has proven to be versatile and efficient. Well-known approaches like the pattern search algorithm were mathematically proven to converge [11]. A cooperative approach of a

* Corresponding author.
E-mail address: b.hofmeister@isd.uni-hannover.de (B. Hofmeister).

https://doi.org/10.1016/j.engstruct.2019.05.047
Received 10 December 2018; Received in revised form 13 May 2019; Accepted 16 May 2019

metaheuristic and a deterministic optimisation algorithm was introduced in [12] and further developed in [13], which resulted in an improvement of the convergence rate. We introduce global pattern search as a novel approach in the class of deterministic, derivative-free, global algorithms. Based upon the well-known pattern search algorithm, this global extension features a simple parameterisation by only one parameter as well as high convergence rates. We present results for established test functions and consider aspects of the numerical performance and parallelisation.

Engineering problems such as finite element model updating or design optimisation often constitute an objective value space with multiple local minima. Local optimisers thus converge locally and fail to retrieve the globally optimal solution. This can be alleviated by running a local optimiser several times from randomly chosen start vectors. However, this impairs numerical performance, as global convergence can only be warranted for a high number of restarts.

A typical engineering application for derivative-free global algorithms is the updating of finite element (FE) models. The main idea of model updating in structural engineering is to correct invalid assumptions that are inevitably made when creating a numerical model of a parameterised structure [14,15]. Another application is the identification of changes in the structural behaviour in order to locate and also quantify damage in terms of stiffness and mass deviations [16]. The model updating process is based on objective functions, which capture deviations, e.g., in dynamic properties. Therefore, the FE model is parameterised and the design variables are mapped onto structural properties. The objective function compares the dynamic behaviour of the model to a target state, whereby the optimisation algorithm finds a model to match this target state. In this work, we update the mechanical stiffness of a wind turbine rotor blade in order to localise a structural defect. Commonly, metaheuristic optimisation algorithms are used for model updating [17–19]. Other approaches make use of local optimisation algorithms [20] or a combination of a global and a local optimiser [21,22]. In this contribution, for the first time, we introduce the proposed global pattern search approach to an FE model updating problem.

2. Global pattern search algorithm

One of the oldest and simplest derivative-free deterministic optimisation algorithms is the coordinate descent algorithm, for which a recent review can be found in [23]. Another well-known approach is the direct search algorithm [24]. Both of these algorithms employ a round-robin variation of each variable around a base point in the design variable space. The aforementioned algorithms are local optimisation algorithms and thus not well-suited for problems with multiple local minima. The basic idea of the round-robin variation using a defined step width is expanded in this work. On the one hand, the search strategy is globalised by adding a ‘hall of fame’, which contains the best samples so far. On the other hand, a caching strategy significantly reduces the number of objective function evaluations by eliminating redundant sampling locations.

We solve the scalar, bounded, unconstrained, non-linear and derivative-free optimisation problem

$$\text{minimise } f(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in \mathbb{R}^{n}, \tag{1}$$

where f is the scalar objective function and the vector $\boldsymbol{x}$ comprises n design variables. The space of the design variables is bounded to the volume of a hypercube

$$\boldsymbol{x}_{\mathrm{lb}} \leqslant \boldsymbol{x} \leqslant \boldsymbol{x}_{\mathrm{ub}}, \tag{2}$$

where $\boldsymbol{x}_{\mathrm{lb}}$ and $\boldsymbol{x}_{\mathrm{ub}}$ are the lower and upper bounding vectors. In this work, we do not consider any constraints to the optimisation problem.

The pattern search algorithm [25], which is another well-established local derivative-free optimisation algorithm, employs iterative subdivisions of the search interval. If the step width is exactly halved each time, the resulting search pattern is located on a grid with a resolution of 2^N, where N denotes the number of subdivisions made by the algorithm. Thus, the proposed algorithm expresses the continuous design variable space by integer coordinates, given the parameter N. The repetitive use of a subdivision scheme results in self-similar patterns in the design variable space.

The integer coordinates are mapped onto the continuous design variable space by a linear transformation

$$x_i = x_{\mathrm{lb},i} + s_i \left(x_{\mathrm{ub},i} - x_{\mathrm{lb},i}\right) 2^{-N}, \tag{3}$$

where $\boldsymbol{s}$ denotes the design variable vector represented on the integer grid and i denotes the design variable index. Due to this integer-based approach, algorithms based on grids can handle integer design variables without modifications. In engineering tasks, the precision requirements on the location of the optimum are often quite low. It is rather more important to actually find the global optimum and not to converge to an undesirable local optimum. For example, if the range of a design variable is 1 m and it is discretised using $N = 10$, the resolution is already less than 1 mm. Usually, $N \leqslant 10$ is thus sufficient to yield an effective solution in practical applications. Furthermore, in design optimisation of structures, the choice of measurements is often stipulated by design standards. A prominent example are tube diameter and wall thickness, where only discrete dimensions are available. Integer design variables such as these are readily handled by the proposed algorithm, because it is grid-based.

To achieve a global optimisation, the proposed algorithm tracks a number of T globally best coordinates. This can be interpreted much like the population size in a metaheuristic algorithm: Small populations tend to local convergence with few objective function evaluations, whereas large populations tend to explore the design variable space more globally, but also lead to more objective function evaluations. In each iteration, the list of globally best coordinates - the so-called ‘hall of fame’ $\boldsymbol{H}$ - is updated and sampling is continued using the current set of best coordinates. The first sampling coordinate in this scheme is always located at the centre of the design variable space. Subsequent sampling points are used to fill up the ‘hall of fame’, until it reaches the size T. Thereafter, only the T globally best coordinates are kept in the ‘hall of fame’.

This extension of the pattern search algorithm leads to a generalisation: For $T = 1$, a local search algorithm very similar to pattern search [25] is recovered, for $T \to \infty$, the global grid search approach is recovered, which simply samples a uniform grid in the design variable space. The algorithm is thus dubbed ‘global pattern search’.

When the objective function is evaluated for a coordinate $\boldsymbol{x}$, the result $y = f(\boldsymbol{x})$ is stored in a cache which prevents redundant evaluations of the objective function. The use of integer coordinates for the ‘hall of fame’ avoids floating-point precision issues and makes caching based on coordinates easy. If the algorithm advances twice into one direction, it will sample grid points visited previously. The already visited sample points will then be removed by the caching mechanism. A hash-based dictionary data structure is used for caching to achieve optimal performance.

Using the step width vector $\boldsymbol{w}$, a positive and a negative variation along each design space dimension is calculated for the coordinates in the ‘hall of fame’. These resulting vectors are stored in a matrix containing T samples for the negative and positive variations, respectively.

$$s_{j,i} = b_i + \delta_{ij} w_i, \qquad s_{nT+j,i} = b_i - \delta_{ij} w_i, \tag{4}$$

where $\boldsymbol{b}$ is a base vector, $\boldsymbol{w}$ denotes the step width vector and the indices i and j denote the design variable and sample number, respectively. The Kronecker function δ_{ij} is used to perform the one-at-a-time variation. In total, these expressions generate $2T$ sampling locations for each iteration of the algorithm, which leads to a cross pattern in the design variable space.

The solution is refined one more time, if a new optimum is found. Using the updated 'hall of fame', the cross pattern variation is performed again. Due to the caching scheme, redundant sampling of previously visited points is suppressed and does not affect the algorithm's performance. If the solution cannot be improved further, the step width w_i is halved and the algorithm progresses. The one-at-a-time variation is sufficient and no performance improvements were observed, when combinations of coordinates were taken into account.

The proposed global pattern search scheme is shown in Algorithm 1. The caching as well as the 'hall of fame' work with integer coordinates, while the evaluation of the objective function takes place in $\mathbb{R}^n$.

Algorithm 1. Global pattern search

$w_i \leftarrow 2^{N-1}$ {initialise step width vector}
$H_{1,i} \leftarrow 2^{N-1}$ {initialise hall of fame with centre of design variable space}
loop
  $\boldsymbol{b}_k \leftarrow \boldsymbol{H}_k$ {take T base coordinates from hall of fame}
  for $k = 1 \rightarrow T$ **do**
    generate $2n$ sampling coordinates $\boldsymbol{s}_j$ for each $\boldsymbol{b}_k$
    clamp sampling coordinates $\boldsymbol{s}_j$ to design variable space $[0: 2^N]$
    deduplicate $\boldsymbol{s}_j$ using cache
    calculate $\boldsymbol{x}_j$ from $\boldsymbol{s}_j$
    $y_j \leftarrow f(\boldsymbol{x}_j)$ {sampling}
  end for
  update hall of fame $\boldsymbol{H}$ using $\boldsymbol{y}$
  if new global optimum found, **then**
    continue loop
  end if
  if every w_i **is** 1, **then**
    break loop
  end if
  $w_{max} \leftarrow w_{max} \backslash 2$ {reduce largest step width}
end loop

During run-time, the algorithm needs to store the coordinates of every sampled point to enable the caching. Objective values and coordinates for samples that are in the 'hall of fame' need to be stored as well. Objective values for coordinates not listed in the 'hall of fame' do not need to be stored. The update is facilitated by replacing the highest objective value entry by a sampling point with a lower objective value.

For arbitrarily sized design variable spaces, the step width still has to be a power of two, so that the caching strategy can efficiently reduce the number of samples. Sampling on the upper boundary of the design variable space needs to be treated specially in order that the generated sampling points can snap back into the regular grid. The lower boundary is always located at the origin of the coordinate system, thus being always situated on the regular grid. This feature can be achieved by integer arithmetic, where \ denotes truncating division. The coordinates $\boldsymbol{H}_k$ taken from the 'hall of fame' are transformed before being used as base coordinates $\boldsymbol{b}_k$ in the variation

$$b_{k,i} = w_i(H_{k,i} \backslash w_i). \tag{5}$$

2.1. Parallelism and depth-first search

In most engineering optimisation problems, the evaluation of the objective function is computationally very expensive, in some cases taking several hours. In order to speed up the optimisation procedure, multiple samples are usually evaluated in parallel. In the proposed algorithm, the sampling can readily be carried out in parallel, since the samples $\boldsymbol{x}_j$ are generated directly in vector format. Algorithm 1 can be interpreted as a breadth-first search. With the proposed global pattern search approach, it is possible to boost the convergence rate by trading off the aforementioned parallelism properties, which will speed up the search for the global optimum.

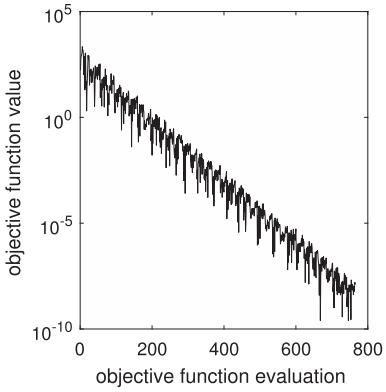

Fig. 1. Objective value history for Himmelblau function using Algorithm 1, $N = 20$, $T = 10$.

Algorithm 2. Depth-first search

initialise sample cache
for $T = 1$ **to** $T_{\max}$ **do**
  run Algorithm 1
end for

In the scheme shown in Algorithm 2, the parameter T denoting the number of tracked globally best coordinates is incremented starting from one and the algorithm is restarted repetitively. In this mode, the sample cache also needs to retain objective function values throughout the process. In the first iteration with $T = 1$, a local search algorithm is recovered. With n being the number of design variables, there are at most $2n$ samples generated at once, so that the performance gain by parallel objective function evaluations is low. As T is increased, the search pattern emphasises global exploration, until the last iteration using $T_{\max}$ recovers the same result as Algorithm 1.

Fig. 2 shows that initially, the convergence of the depth-first search is rapid compared to the breadth-first approach depicted in Fig. 1. With the parameter T starting at 1, a local optimisation is performed at first. One of the four minima of the Himmelblau function [24] is discovered with good numerical precision within less than 100 function evaluations. With increasing parameter T, small improvements of the objective function value are achieved and the other global minima are discovered. Thereby, the proposed depth-first scheme is only effective, if the resolution parameter N is chosen reasonably small. If this is the case, Algorithm 1 terminates sufficiently early for the parameter T to increment. Thus, global convergence can be achieved with a low number of objective function evaluations. With $T_{\max} \rightarrow \infty$ and T incrementing gradually, the design space is explored with sampling points

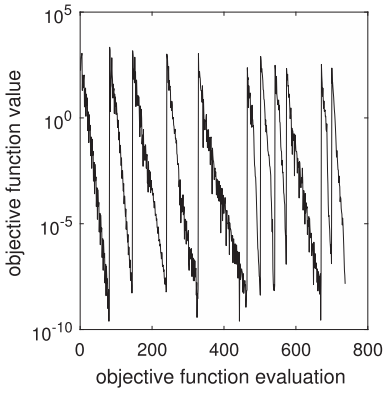

Fig. 2. Objective value history for Himmelblau function using Algorithm 2, $N = 20$, $T_{max} = 10$.

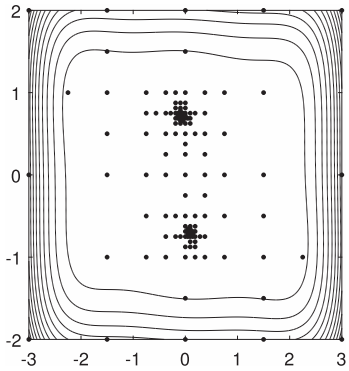

Fig. 3. Camel6 function [26], $N = 20$, $T = 5$, 415 evaluations.

even further away from the optima, until finally grid search is recovered.

In iterations with a low parameter T, the depth-first approach will converge on other local minima than in subsequent iterations, when T assumes higher values. As more samples are generated around these local minima, this generally results in more objective function evaluations to reach the same optimum, when compared to Algorithm 1. This can be interpreted as a trade-off between quick initial convergence and total number of function evaluations.

2.2. Benchmark tests

Figs. 3–8 show sampling patterns obtained by applying Algorithm 1 to various two-dimensional benchmark problems. The corresponding objective value space is visualised using contour lines. Generally, a self-similar search pattern appears in the design variable space near the optima as the optimisation converges. The parameter T was adjusted to the respective problem for robust convergence while maintaining a low number of objective function evaluations to illustrate best case performance. We use $N = 20$ for all of these problems to demonstrate the convergence rate of the algorithm. For practical optimisation problems, a lower resolution would suffice. For each test function, we record the number of objective function evaluations needed until the algorithm terminates.

The non-uniform scaling of the step width improves the performance of the algorithm on smooth problems. As can be seen in the sampling of the benchmark functions, refinement on only one axis is often sufficient to improve the corresponding objective function value. Due to the unbiased tracking of local optima, the proposed deterministic approach does not converge on a single global optimum. Instead, if there are several optima with the same objective value, the sampling density in these regions is roughly the same. This becomes evident looking at the performance of the algorithm on the test functions Camel6 [26] and Himmelblau [24]. Beyond that, the distribution of

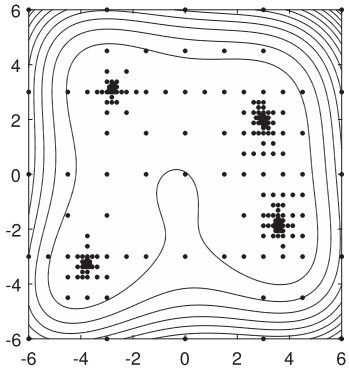

Fig. 4. Himmelblau function [24], $N = 20$, $T = 10$, 765 evaluations.

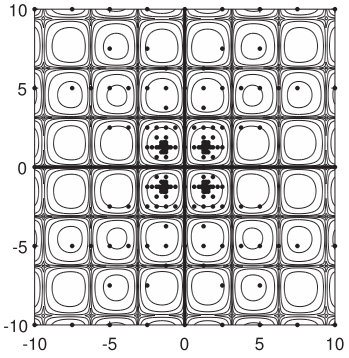

Fig. 5. Cross-in-tray function [27], $N = 20$, $T = 8$, 681 evaluations.

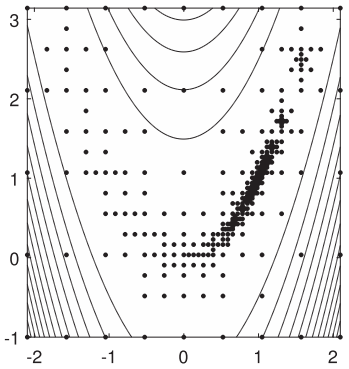

Fig. 6. Rosenbrock function [28], $N = 20$, $T = 15$, 1171 evaluations.

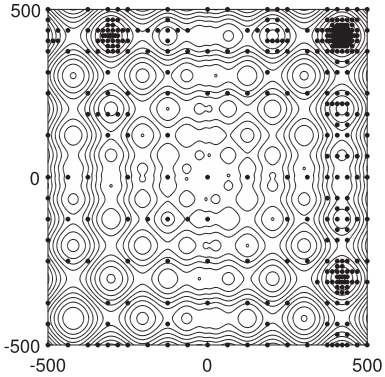

Fig. 7. Schwefel function 128 [29], $N = 20$, $T = 20$, 1059 evaluations.

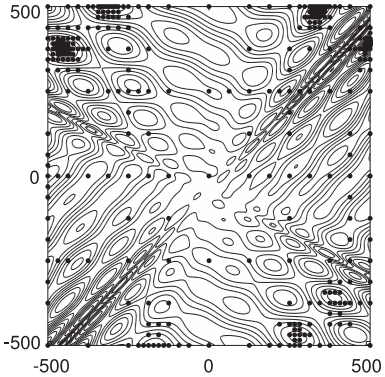

Fig. 8. Eggholder function [27], $N = 20$, $T = 20$, 955 evaluations.

sampling points in the objective value space of the Cross-in-tray [27] test function displays the symmetry properties of the unbiased tracking: The sampling pattern reflects the symmetry of this test function about both axes. We intentionally chose the parameter T to be a multiple of four in this case. As a consequence, the numbers of 'hall of fame' entries in each global optimum are the same, leading to the evident symmetric sampling pattern.

The described unbiased tracking property is conducive to cluster analysis, e.g., by using an agglomerate hierarchical approach. This way, up to T distant near-optimal points in the design variable space can be identified with only one optimisation run. In an engineering optimisation task, the near-optimal design solutions obtained this way may even contain a better solution than the actual global optimum. Such cases arise when there are other objectives or design aspects that are not considered by the objective function. Classical metaheuristic algorithms converge on only one minimum, restarts are thus required to identify all global minima. This issue is addressed by multi-modal variants of metaheuristic algorithms [30]. In this regard, the convergence characteristic of the proposed algorithm provides a valuable tool.

3. Finite element model updating

For the application of the proposed optimisation algorithm on a typical engineering problem, we choose the model updating of an offshore wind turbine rotor blade. This particular application addresses the identification of a structural defect due to changes in the structural behaviour.

We create a parameterised FE model of a typical offshore wind turbine blade. The model of the healthy structure represents the initial model of the updating process and is called the reference model throughout this work. Since we have no access to measurement data of damaged blades, we create an additional model with a fictitious damage, subsequently called the target state. The fictitious damage is simulated by reducing the stiffness of certain blade sections. To locate this artificially induced change in the structural behaviour, the stiffness parameters of the reference model are adapted to match the target state by comparing modal properties. In this case, the objective function comprises mode shapes. Sections with reduced stiffness then indicate the area where damage has occurred.

Since the presented work includes the comparison of the applicability as well as the efficiency of the proposed global pattern search algorithm with other well-established optimisers, the computational costs of multiply evaluating the numerical model become an issue. Therefore, we choose a very simple numerical model of the blade consisting of beam elements, because it is computationally inexpensive. A detailed description of the FE model as well as the specification of the design variables and the objective function are given in the following subsections.

3.1. Model description

The NREL offshore 5-MW baseline wind turbine [31] is a widely used model of a typical multi-megawatt wind energy converter. Its detailed publicly accessible specifications are used to create an FE model of one of its blades. The simulations are conducted using the FE analysis software ABAQUS. In order to reduce the computational costs for calculating the objective function value, the 63-metre long blade is modelled as a beam model consisting of 49 elements. The sectional blade properties listed in [31] are assigned to the elements using a general section definition.

To create a realistic experimental test setup of a wind turbine blade, standard blade tests are considered. Blade tests consist of periodic load applications in order to trigger failure by fatigue as well as applications of non-periodic loads such as impact, static or dynamic loads [32]. In these tests, the blades are fixed at the hub end and the loads are applied by clamps which are attached to a specific section. In the numerical model of the NREL blade, all degrees of freedom are fixed at the hub end and two clamps are simulated at 21 m and 42 m from the hub. The weight of the clamps is set to 1500 and 500 kilogrammes, respectively. This beam model represents the reference model of the healthy wind turbine blade. We additionally create the target state, assuming that fatigue damage has occurred. To mimic the fatigue state, the edgewise bending stiffness EI_{edge} of certain elements along a predefined length of the blade is reduced. For the purpose of locating this imitated damage, the mode shapes of the target state are calculated, representing the feature normally extracted from the measured response. Since a realistic measurement can only be carried out with a limited number of sensors, the mode shapes are calculated at a restricted number of predefined locations, representing the sensor positions.

3.2. Design variables

The choice of design variables is significant, because they have a great impact on the quality of the optimisation procedure and its result. Thereby, the determination of the design variables strongly depends on the purpose of model updating [14].

A common approach to locate damage is to determine certain regions, where the probability of an emerging defect is known to be high, based on experience, prior knowledge or error localisation methods [21,33]. Thereby, the design variables typically represent scaling factors for the stiffness of certain FEs in these regions. If there exist many regions where stiffness is assumed to be subject to change, the amount of design variables and thus, the dimension of the optimisation problem increases. This usually results in an objective value space with many local minima which can make numerical optimisation unfeasible [34]. Additionally, oscillatory stiffness values might produce nearly the same response as the targeted one, despite being physically unrealistic. Therefore, the amount of design variables should be kept low.

Several approaches address this problem, most commonly used is the assignment of one design variable to a group of FEs supposedly having similar mechanical properties [17,33]. These groups are called substructures or super-elements. In this application, we are pursuing an alternative approach. Instead of directly using design variables that represent stiffness scaling factors for certain, predefined FEs or substructures, we introduce a damage distribution function that can be described by a few parameters only. A similar approach, which is motivated by smoothly distributed structural properties, is analysed and successfully used in [35], whereby a quadratic function is used to describe a damage function. Fig. 9 illustrates a damage distribution function designed for beam structures that is described by the design variable vector

$$\mathbf{x} = \begin{pmatrix} \mu \\ D \\ \sigma \end{pmatrix}. \tag{6}$$

In this vector, D represents the intensity of the damage, the mean value μ represents the geometrical position of the damage distribution's centre point along the blade and the standard deviation σ represents the width of the distribution. The points along the leading edge of the blade

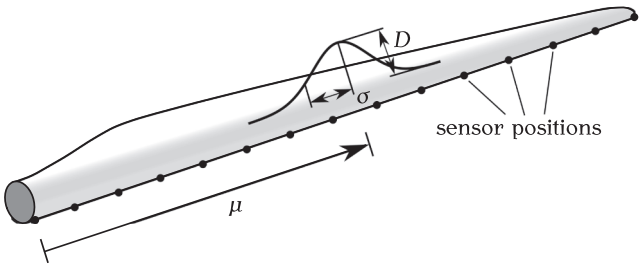

Fig. 9. Parameterisation of damage distribution and sensor positions along the blade length.

depict the sensor positions, which are placed every 4 m along the blade length. By employing this damage distribution function, the proposed parameterisation is independent of the FE discretisation as well as of prior assumptions about the defect location. This is an important advantage.

The damage intensity can be described as

$$D = \int_L 1 - \theta(s)\, ds, \tag{7}$$

where L is the total blade length, s is the control variable along the blade length and $\theta(s)$ is the stiffness scaling factor at position s. Since the beam model of the wind turbine blade is designed with a discrete amount of elements, there also exists a discrete number of stiffness scaling factors θ_i that are calculated at the geometrical centre s_i of each element. Thus, the integral can be expressed as the sum over the total number of N_{el} blade elements

$$D = \sum_{i=1}^{N_{\text{el}}} 1 - \theta_i. \tag{8}$$

In this work, we assume a damage that is normally distributed along the longitudinal axis of the blade. This can be expressed using a cumulative distribution function $F(s, \mu, \sigma)$, resulting in the following equation for calculating the stiffness scaling factors for each element

$$\theta_i = 1 - \left(D \cdot F(s_i, \mu, \sigma) - \sum_{j=1}^{i-1} 1 - \theta_j \right), \tag{9}$$

where, again, s_i describes the geometrical centre of each element along the blade length, starting from the hub. This means that the initial stiffness EI^0 of each element is reduced by the corresponding scaling factor θ_i, which is calculated at the geometrical centres of the elements as described by

$$EI_i^{\theta} = \theta_i \cdot EI_i^0. \tag{10}$$

The target state is also simulated by applying the described cumulative damage distribution function using a fictitious damage position, a fictitious intensity as well as a fictitious width.

3.3. Objective function

Besides the choice of the design variables, the formulation of the objective function is another important factor in the process of model updating. The objective function numerically compares the experimental results with the results of the FE analysis. For this reason, it has to be sensitive even to small changes in the structural behaviour [33,18].

The features compared by the objective function can be defined in modal, frequency or time domain. In this paper, the model updating method is based on mode shapes, since these dynamic features can be obtained experimentally in high quality [36]. Only the first four mode shapes with a significant amplitude in edge direction are considered. To evaluate the correlation between the relevant mode shapes, the modal assurance criterion [37] is calculated. This criterion determines the degree of similarity between the two mode shape vectors Φ_t and Φ_s

$$\text{MAC}_i(\theta) = \frac{(\Phi_{t,i}^T \Phi_{s,i}(\theta))^2}{(\Phi_{t,i}^T \Phi_{t,i})(\Phi_{s,i}^T(\theta) \Phi_{s,i}(\theta))}, \tag{11}$$

where i denotes the mode number and the indices t and s denote target and simulated quantities, respectively. The modal assurance criterion returns a value of one, if the compared mode shapes are linearly dependent and a value of zero, if they are linearly independent.

A common method to compute the error between numerical and experimental data is the least squares approach [38]. Considering N_f eigenfrequencies, we formulate the objective function

Table 1
Design variables chosen for the simulation of the target state and their permissible range.

Design variables	Fictitious damage	Bounds	
		$\mathbf{x_{lb}}$	$\mathbf{x_{ub}}$
μ in m	15	0	63
D	0.02	−0.05	0.05
σ in m	2	–	–

$$\rho(\theta) = \sum_{i=1}^{N_f} (1 - \text{MAC}_i(\theta))^2. \tag{12}$$

Since we use the same method for calculating the fictitious stiffness reduction as for calculating the stiffness scaling factors in our model updating scheme, the objective function value becomes zero when the correct design variables are evaluated.

4. Results

For the application of the proposed algorithm, a two-dimensional model updating problem is considered. We choose the mean value and the damage intensity of the previously described damage distribution as the design variables

$$\mathbf{x} = \begin{pmatrix} \mu \\ D \end{pmatrix}. \tag{13}$$

The standard deviation is set to the constant value $\sigma = 2$ m, which is similar in size to experimentally observed blade damage patterns [39]. Table 1 lists the variables describing the cumulative damage distribution function used for creating the fictitious defect. Additionally, the permissible ranges, determining the upper and lower bounds, are given.

Fig. 10 illustrates the cumulative damage distribution function used for the simulation of the target state. Additionally, Fig. 10 shows the corresponding stiffness scaling factors calculated for each element by plugging the values listed in Table 1 in Eq. 9. The objective value space of this two-dimensional optimisation problem is given in Fig. 11. Thereby, a positive damage factor is equatable to the loss of stiffness, whereas a negative damage factor means stiffening. The global minimum emerges as a prominent peak in the design variable space, whereby its location is defined by the design variables given in Table 1. At this location, the objective function value is equal to zero. However, regarding Fig. 11, it is apparent that the objective function additionally possesses multiple local minima as well as a waviness which are caused by the mode shapes it is based upon.

The optimisation result of the proposed global pattern search algorithm is obtained using Algorithm 1. An illustration of its performance is shown in Fig. 12, whereby the objective value space is visualised using contour lines. The distribution of the sampling points demonstrates a good balance between global sampling and strong convergence at the optimum. Also, the sampling pattern clearly reflects the self-similar search pattern of the proposed algorithm. In this case, we solved the model updating problem using the parameter $T = 10$. For

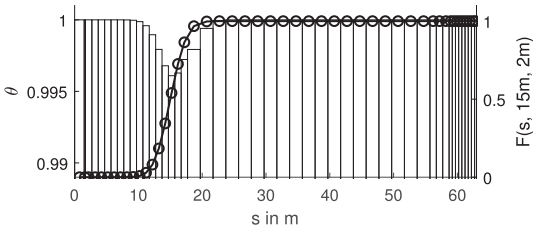

Fig. 10. Cumulative damage distribution function of the fictitious damage and corresponding stiffness scaling factors.

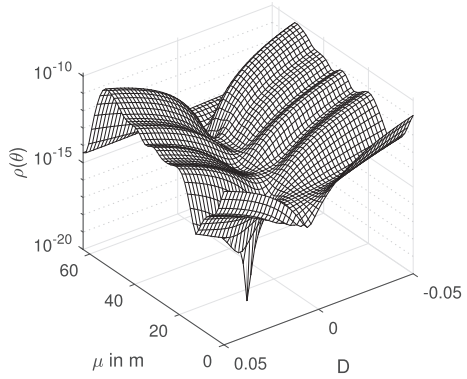

Fig. 11. Objective value space of the two-dimensional optimisation problem.

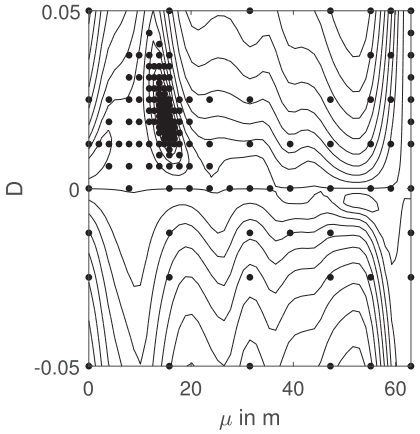

Fig. 12. Objective value space of the two-dimensional model updating problem and samples of the global pattern search algorithm, $T = 10$.

engineering problems with even more local minima in the objective value space, a higher value should be chosen to improve global coverage.

4.1. Comparison of different optimisation algorithms

To validate the efficiency of the proposed global pattern search algorithm, its performance is compared to the performance of two metaheuristic and one local optimisation algorithm. Therefore, the following well-known and established algorithms are chosen: Particle swarm optimisation [3,40], genetic algorithm [2] as well as sequential quadratic programming [1] with randomised start vectors.

Hybrid optimisation schemes, consisting of a global and a subsequent local optimisation algorithm, can provide a good performance on model updating problems [22]. The complexity arising when parameterising a hybrid approach have been pointed out by Goldberg et al. [41]. Choices in parameterisation can introduce a bias, which increases the performance on certain optimisation problems, while deteriorating performance on others [42]. To illustrate distinctive numerical features of different algorithms, we thus focus on basic optimisation schemes of global and local optimisers in our comparison.

The algorithms are applied in their original state as described in the corresponding literature and the parameters are set to values recommended by their respective original authors. For the sake of comparability, the maximum number of objective function evaluations is set to 500 for all algorithms. We obtain statistical parameters for the minimum objective value by running the optimisation 100 times with each optimiser. As this particular model updating problem is based on an FE code with limited numerical precision, the objective function is discrete at small scales. This poses numerical restrictions to the minimum finite difference step size and can lead to premature triggering of the termination condition in derivative-based optimisers. Thus, the number of objective function evaluations of the sequential quadratic programming algorithm matches not always the allowed maximum of 500 evaluations. The design variable space is normalised by

$$\mathbf{x}_{scaled} = \frac{\mathbf{x} - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}, \tag{14}$$

which is necessary for the application of the described algorithms to avoid convergence issues due to anisotropy. Because metaheuristic algorithms explore the design variable space stochastically, every run yields a different result. The same applies to the application of local optimisation algorithms when started from randomly chosen start vectors. On the contrary, the proposed algorithm always converges to the same result because of its deterministic nature. This means that the standard deviation is always zero regardless of the number of runs, provided that the parameters are constant. In Table 2, the mean objective function value, its standard deviation and the minimal objective function value reached in the 100 runs are compared. Additionally, the optima of the two metaheuristic and the local optimisation algorithm determined in 100 runs are visualised in Figs. 13–15.

Table 2
Comparison of the results of 100 applications of each optimisation algorithm.

Optimisation algorithm	Objective function value		
	Mean	Standard deviation	Minimum
Particle swarm optimisation	2.07×10^{-17}	1.17×10^{-16}	1.18×10^{-25}
Genetic algorithm	3.50×10^{-18}	2.88×10^{-17}	4.13×10^{-26}
Global pattern search algorithm	1.11×10^{-31}	0	1.11×10^{-31}
Sequential quadratic programming	1.59×10^{-15}	1.75×10^{-15}	7.27×10^{-27}

A comparison of the three contour plots clearly reflects that the metaheuristic algorithms reach the area around the global optimum almost every time. In contrast, the local optimiser converges onto several local minima in addition to the global minimum. The results listed in Table 2 support this observation: The mean objective function value and the standard deviation of the sequential quadratic programming algorithm are orders of magnitude worse than the values obtained using the global optimisers.

With respect to the minimum objective function value reached in 100 runs, the genetic algorithm performs better than the particle swarm optimisation algorithm. The minimum value obtained using sequential quadratic programming is even lower than the ones obtained by the two metaheuristic algorithms, although its overall performance is worse. The objective function value reached by the proposed global pattern

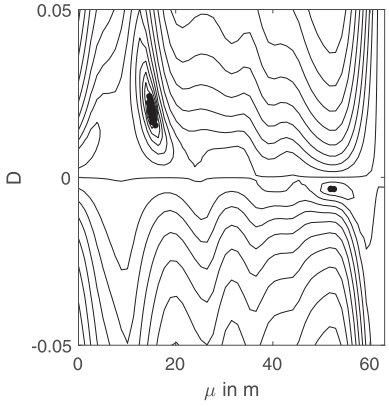

Fig. 13. Particle swarm optimisation.

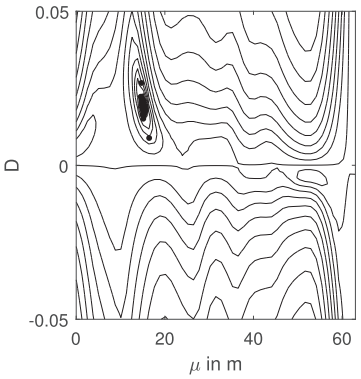

Fig. 14. Genetic algorithm.

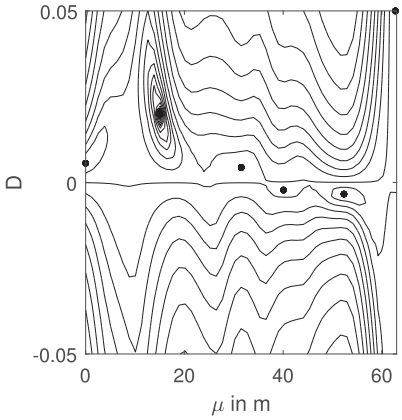

Fig. 15. Sequential quadratic programming.

search algorithm is lower than all minimum values reached by the other algorithms. For the considered model updating problem, this implies that the proposed algorithm reaches the location closest to the global optimum. As the result is also constant and independent of the number of runs, the proposed algorithm outperforms the others.

4.2. Discussion of the results

Due to the strong local convergence of sequential quadratic programming, the minimum objective function value obtained using this algorithm is lower than the ones obtained by the metaheuristic optimisers. But since the trajectory of the local optimiser is dependent on the positioning of the randomly chosen start vector, it often converges to local minima which do not match the global optimum. This explains the low performance of the sequential programming algorithm regarding the mean value and the standard deviation of the residuum. To alleviate this problem, [21] proposed a two-step approach, in which a local optimiser is started from the point of convergence of a metaheuristic algorithm.

Due to the previously described numerical restrictions to the finite difference step size and the resulting premature termination of the local optimiser, the implementation of the sequential quadratic programming algorithm referred to as “fmincon’ [43] was not able to obtain an objective function value as low as the one obtained by the proposed global pattern search approach. Thus, we conclude that the proposed optimisation algorithm performs better on the two-dimensional model updating problem than the well-established optimisation algorithms chosen for comparison. It is especially suited for computationally expensive optimisation problems, since no restarts are required to obtain global convergence. This reduces the computational costs extremely while also providing quick and robust convergence.

A potential drawback of the proposed method is the determination of the parameter T, which controls the number of tracked sampling points. The convergence as well as the numerical performance of the global pattern search algorithm largely depend on this numerical value. If it is chosen too low, the algorithm will converge locally and thus fail to yield a globally optimal result.If it is chosen too high, the resulting performance in terms of convergence rate will be low. The value for T cannot be determined analytically. This means that numerical studies are needed to find a value, which is suitable for the optimisation problem at hand.

5. Summary and outlook

In this work, we demonstrated that the proposed global pattern search approach can efficiently solve test problems as well as the model updating problem. A comparison of the performance of the proposed method to established metaheuristic and local approaches was conducted for a number of 100 objective function evaluations. We found that the proposed method converges quicker and also finds the global optimum more robustly. Since metaheuristic optimisers need a high amount of objective function evaluations due to their probabilistic search pattern and derivative-based local optimisers require restarts with randomised start vectors, both approaches are computationally expensive. On the contrary, the proposed method extremely reduces the computational costs due to its quick and unique convergence regardless of the number of runs. This is why the introduced global pattern search approach shows great potential for engineering applications like FE model updating. Furthermore, the proposed algorithm can identify multiple global optima in one run as shown with the test functions Himmelblau, Camel6 and Cross-in-tray.

Throughout this work, we considered single-objective optimisation problems only. However, the caching approach proposed in this paper also features interesting properties for multi-objective optimisation problems. Furthermore, an extension of the introduced damage distribution function to a three-dimensional structure provides a more accurate damage localisation. Due to the promising performance of the proposed algorithm on the model updating problem, we conclude that deterministic optimisation methods should receive more attention in the field of engineering optimisation.

Acknowledgments

We greatly acknowledge the financial support of the German Federal Ministry for Economic Affairs and Energy (research projects *Deutsche Forschungsplattform für Windenergie*, FKZ 0325936E and Multivariates Schadensmonitoring von Rotorblättern, FKZ 0324157A) that enabled this work.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.engstruct.2019.05.047.

References

[1] Han S-P. A globally convergent method for nonlinear programming. J Optim Theory Appl 1977;22(3):297–309. https://doi.org/10.1007/BF00932858.
[2] Goldberg DE. Genetic algorithms in search, optimization and machine learning. 1st Edition Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1989.
[3] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN’95 – International Conference on Neural Networks, vol. 4; 1995. p. 1942–8. doi:https://doi.org/10.1109/ICNN.1995.488968.
[4] Geem ZW, Kim JH, Loganathan G. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8. https://doi.org/10.1177/003754970107600201.
[5] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.
[6] Dasgupta S, Das S, Abraham A, Biswas A. Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. IEEE Trans Evol Comput 2009;13(4):919–41. https://doi.org/10.1109/TEVC.2009.2021982.

[7] Shayeghi H, Dadashpour J. Anarchic society optimization based pid control of an automatic voltage regulator (avr) system. Electric Electron Eng 2012;2(4):199–207. https://doi.org/10.5923/j.eee.20120204.05.

[8] Cuevas E, Cienfuegos M. A new algorithm inspired in the behavior of the social-spider for constrained optimization. Expert Syst Appl 2014;41(2):412–25. https://doi.org/10.1016/j.eswa.2013.07.067.

[9] Rios LM, Sahinidis NV. Derivative-free optimization: a review of algorithms and comparison of software implementations. J Global Optim 2013;56(3):1247–93. https://doi.org/10.1007/s10898-012-9951-y.

[10] Jones DR, Perttunen CD, Stuckman BE. Lipschitzian optimization without the lipschitz constant. J Optim Theory Appl 1993;79(1):157–81. https://doi.org/10.1007/BF00941892.

[11] Dolan E, Lewis R, Torczon V. On the local convergence of pattern search. SIAM J Optim 2003;14(2):567–83. https://doi.org/10.1137/S1052623400374495.

[12] Sotiropoulos DG, Stravopoulos EC, Vrahatis MN. A new hybrid genetic algorithm for global optimization. In: Proceedings of the 2nd world congress of nonlinear analysis, vol. 30; 1997. p. 4529–38. doi:https://doi.org/10.1016/S0362-546X(96)00367-7.

[13] Alliot J-M, Durand N, Gianazza D, Gotteland J-B. Finding and proving the optimum: cooperative stochastic and deterministic search. In: ECAI; 2012. p. 55–60. doi:https://doi.org/10.3233/978-1-61499-098-7-55.

[14] Friswell MI, Mottershead JE. Finite element model updating in structural dynamics. Solid mechanics and its applications vol. 38. Netherlands: Springer; 1995. https://doi.org/10.1007/978-94-015-8508-8.

[15] Mottershead JE, Friswell MI. Model updating in structural dynamics: a survey. J Sound Vib 1993;167(2):347–75. https://doi.org/10.1006/jsvi.1993.1340.

[16] Fan W, Qiao P. Vibration-based damage identification methods: a review and comparative study. Struct Health Monitor 2011;10(1):83–111. https://doi.org/10.1177/1475921710365419.

[17] Levin R, Lieven N. Dynamic finite element model updating using simulated annealing and genetic algorithms. Mech Syst Signal Process 1998;12(1):91–120. https://doi.org/10.1006/mssp.1996.0136.

[18] Jahjouh MM. A modified adaptive harmony search algorithm approach on structural identification and damage detection. Hannover: Institut für Baumechanik und Numerische Mechanik, Gottfried Wilhelm Leibniz Universität Hannover; 2016.

[19] Jahjouh MM, Nackenhorst U. a modified harmony search approach on structural identification and damage detection of wind turbine supporting structures. J Vibroengineering 1876;18(1). <https://www.jvejournals.com/article/16694>.

[20] Petersen ØW, Øiseth O. Sensitivity-based finite element model updating of a pontoon bridge. Eng Struct 2017;150:573–84. https://doi.org/10.1016/j.engstruct.2017.07.025.

[21] Schröder K, Gebhardt CG, Rolfes R. A two-step approach to damage localization at supporting structures of offshore wind turbines. Struct Health Monitor 2018;17(5):1313–30. https://doi.org/10.1177/1475921717741083.

[22] Begambre O, Laier J. A hybrid Particle Swarm Optimization – Simplex algorithm (PSOS) for structural damage identification. Adv Eng Softw 2009;40(9):883–91. https://doi.org/10.1016/j.advengsoft.2009.01.004.

[23] Wright SJ. Coordinate descent algorithms. Math Program 2015;151(1):3–34. https://doi.org/10.1007/s10107-015-0892-3.

[24] Himmelblau DM. Applied nonlinear programming. McGraw-Hill; 1972.

[25] Hooke R, Jeeves TA. direct search solution of numerical and statistical problems. J ACM 1961;8(2):212–29. https://doi.org/10.1145/321062.321069.

[26] Molga M, Smutnicki C. Test functions for optimization needs, Tech. rep., Department of Automation, Mechatronics and Control Systems, Polytechnika Wrocławska; 2005. <http://new.zsd.iiar.pwr.wroc.pl/files/docs/functions.pdf>.

[27] Jamil M, Yang X-S. A literature survey of benchmark functions for global optimisation problems. Int J Math Modell Numer Optim 2013;4(2):150–94. https://doi.org/10.1504/IJMMNO.2013.055204.

[28] Rosenbrock H. An automatic method for finding the greatest or least value of a function. Comput J 1960;3(3):175–84. https://doi.org/10.1093/comjnl/3.3.175.

[29] Schwefel H-P. Numerical optimization of computer models. John Wiley & Sons Inc; 1981.

[30] Goldberg DE, Richardson J, et al., Genetic algorithms with sharing for multimodal function optimization. In: Genetic algorithms and their applications: proceedings of the second international conference on genetic algorithms, Hillsdale, NJ: Lawrence Erlbaum; 1987. p. 41–9.

[31] Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-MW reference wind turbine for offshore system development, Tech. Rep. NREL/TP-500-38060, 947422, National Renewable Energy Laboratory; 2009. doi:https://doi.org/10.2172/947422.

[32] Yang B, Sun D. Testing, inspecting and monitoring technologies for wind turbine blades: a survey. Renew Sustain Energy Rev 2013;22:515–26. https://doi.org/10.1016/j.rser.2012.12.056.

[33] Kim G-H, Park Y-S. An improved updating parameter selection method and finite element model update using multiobjective optimisation technique. Mech Syst Signal Process 2004;18(1):59–78. https://doi.org/10.1016/S0888-3270(03)00042-6.

[34] Bruns M, Hofmeister B, Pache D, Rolfes R. Finite element model updating of a wind turbine blade—a comparative study. EngOpt 2018 proceedings of the 6th international conference on engineering optimization Springer International Publishing; 2019. p. 569–80. https://doi.org/10.1007/978-3-319-97773-7_51.

[35] Teughels A, De Roeck G, Suykens JA. Global optimization by coupled local minimizers and its application to FE model updating. Comput Struct 2003;81(24–25):2337–51.

[36] Boeswaldsa M, Goege D, Fuellekrug U, Govers Y. A review of experimental modal analysis methods with respect to their applicability to test data of large aircraft structures. Proc. of the international conference on noise and vibration engineering ISMA. 2006. p. 2461–82.

[37] Allemang RJ, Brown DL. A correlation coefficient for modal vector analysis. In: Proceedings of the 1st international modal analysis conference, vol. 1, SEM Orlando; 1982. p. 110–6.

[38] Link M. Updating of analytical models – review of numerical procedures and application aspects. In: Proc., structural dynamics forum SD2000. Baldock, UK: Research Studies Press; 1999. p. 193–223.

[39] Tsiapoki S, Häckell MW, Grießmann T, Rolfes R. Damage and ice detection on wind turbine rotor blades using a three-tier modular structural health monitoring framework. Struct Health Monitor 2018;17(5):1289–312. https://doi.org/10.1177/1475921717732730.

[40] Eberhart R, Shi Yuhui. Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No.01TH8546), vol. 1.Seoul, South Korea: IEEE; 2001. p. 94–100. doi:https://doi.org/10.1109/CEC.2001.934376.

[41] Goldberg DE, Voessner S. Optimizing global-local search hybrids. In: Proceedings of the 1st annual conference on genetic and evolutionary computation-volume 1. Morgan Kaufmann Publishers Inc.; 1999. p. 220–8.

[42] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Trans Evol Comput 1997;1(1):67–82. https://doi.org/10.1109/4235.585893.

[43] MATLAB, version 9.0.0.341360 (R2016a), The MathWorks Inc., Natick, Massachusetts; 2016.