Multi-Objective Global Pattern Search: Effective numerical optimisation in structural dynamics

Benedikt Hofmeister*, Marlene Bruns, Clemens Hübler, Raimund Rolfes

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

Abstract

With this work, a novel derivative-free multi-objective optimisation approach for solving engineering problems is presented. State-of-the-art algorithms usually require numerical experimentation in order to tune the algorithm's multiple parameters to a specific optimisation problem. This issue is effectively tackled by the presented deterministic method which has only a single parameter.

The most popular multi-objective optimisation algorithms are based on pseudo-random numbers and need several parameters to adjust the associated probability distributions. Deterministic methods can overcome this issue but have not attracted much research interest in the past decades and are thus seldom applied in practice. The proposed multi-objective algorithm is an extension of the previously introduced deterministic single-objective Global Pattern Search algorithm. It achieves a thorough recovery of the Pareto frontier by tracking a predefined number of non-dominated samples during the optimisation run. To assess the numerical efficiency of the proposed method, it is compared to the well-established NSGA2 algorithm. Convergence is demonstrated and the numerical performance of the proposed optimiser is discussed on the basis of several analytic test functions. Finally, the optimiser is applied to two structural dynamics problems: transfer function estimation and finite element model updating.

The introduced algorithm performs well on test functions and robustly converges on the considered practical engineering problems. Hence, this deterministic algorithm can be a viable and beneficial alternative to random-number-based approaches in multi-objective engineering optimisation.

Keywords: multi-objective optimisation, pattern search, structural dynamics, model updating

1. Introduction

In engineering optimisation tasks, it is often hard to decide which objective function formulation is best suited to solve the problem at hand. Some objectives can also be conflicting like cost and lifetime of a structure. Furthermore, since engineering optimisation often deals with complex numerical models, there

*Corresponding author
Email address: b.hofmeister@isd.uni-hannover.de (Benedikt Hofmeister)

usually are no derivatives of the objective functions available. Multi-objective derivative-free optimisers are able to solve problems of such nature, and have been studied extensively in the past decades [1].

Contrary to single-objective optimisation problems, multi-objective problems usually possess an infinite number of quasi-optimal, or non-dominated, solutions. Optimisation algorithms for solving such problems thus need to simultaneously provide convergence to all non-dominated solutions. The central concept applied in these algorithms is the Pareto frontier [2], which contains all non-dominated solutions. In order to converge towards the Pareto frontier, feasible points have to be found in the design variable space, which have an improved solution in at least one objective compared to all previously evaluated points.

The most commonly used global derivative-free algorithms for single as well as multi-objective optimisation follow a random-number-based approach. This means, that they employ pseudo-random numbers to generate samples in the design space in order to iteratively minimise the objective functions. Deb et al. presented the multi-objective genetic algorithm NSGA2 [3], which has become a very prominent example of such algorithms. Genetic algorithm variants for multi-objective problems also include micro-GA [4] and Niched Pareto GA [5]. Reviews of similar methods were published by Zitzler [6] and Konak [7]. A more recent branch of meta-heuristic multi-objective optimisers are particle swarm approaches, often referred to as MOPSO [8, 9, 10]. An evolution strategy based algorithm was adapted to multi-objective problems by Beume et al. [11] and its high performance was demonstrated. Among the evolution strategy approaches, there are also PAES [12] and SPEA2 [13], which employ advanced selection criteria in their iteration schemes. More recent evolutionary algorithm approaches include MO-CMA-ES [14], which uses the hypervolume metric in its selection criteria and MOEA/D [15], which decomposes the multi-objective problem into several single-objective problems. Benchmarks including the newer approaches show, that NSGA2 is to date among the most efficient meta-heuristic algorithms [16, 17]. Therefore, an implementation of NSGA2 [18] is considered as the benchmark algorithm for this work.

The aforementioned random-number-based algorithms however share a common drawback in the amount of parameters which govern their convergence properties. As an example, genetic algorithms like NSGA2 often have five parameters. The most important of these parameters is the population size, which influences the balance between convergence rate and exploration. Two further parameters set the behaviour of the genetic crossover operator. The last two parameters influence the mutation rate and distribution. The authors of NSGA2 provide some hints on viable settings for several benchmark problems [19]. However, in practical engineering optimisation, numerical experimentation is required to find satisfactory settings. This general problem inherent to most random-based algorithms has been noted. To this end, the Differential Evolution algorithm [20] was put forward, which features just three parameters. Due to this user-friendly approach, Differential Evolution has since been adopted and improved by many researchers [21]. Nevertheless, the achievable numerical performance suffers due to its lack of tuning parameters. Recently, the criticism of metaphor-based algorithms [22], which all genetic and evolutionary approaches belong to, even led to the

development of an approach [23] that has the population size as its only parameter.

Deterministic algorithms, which do not employ random numbers at all, attract much less attention in the field of global derivative-free optimisation. Yet, some notable work was recently published on deterministic multi-objective approaches. Evtushenko and Posypkin proposed an approach which employs box-constraints and a branch-and-bound optimisation scheme [24]. Al-Dujaili and Suresh demonstrate a method based on the dividing rectangles approach [25]. A hybrid between random-number-based and a pattern search approach was proposed by Custódio [26]. Alotto and Capasso introduced an algorithm based on pattern search and on a hybridisation with meta-heuristic algorithms to improve the performance [27]. All these deterministic and hybrid algorithms have a comparatively low number of parameters in common, often only a single one. However, some of these approaches lack numerical robustness and there have not been many attempts made at comparing them to more mainstream methods like NSGA2.

Another desirable property of optimisation algorithms is the ability to solve multi-objective multi-modal problems. As in single-objective optimisation, the multi-modal problems are characterised by multiple regions in the design variable space which map onto the same objective function values. A review by Tanabe and Ishibuchi [28] shows that evolutionary algorithms dominate this field. The absence of competitive multi-modal deterministic approaches can be attributed to the lack of an effective way to control the balance between convergence rate and exploration inherent to many deterministic optimisation methods. Hence, these algorithms fail to simultaneously converge to multiple areas in the design variable space.

In this paper, a multi-objective variant of the global pattern search algorithm [29] is introduced as a novel approach in multi-objective optimisation, which tackles the above mentioned problems. Results are obtained for established analytic test functions and the features of the proposed method are compared with NSGA2. Both the objective value space as well as the design variable space behaviour are considered and an in-depth assessment of the convergence behaviour is carried out.

Benchmarks in numerical optimisation are carried out on the basis of analytical optimisation test functions. In the past decades, an immense number of these test functions has been proposed by numerous authors. The first multi-objective test functions, which found widespread use, were developed by Schaffer [30]. In the following years, the test functions became more and more sophisticated to showcase the performance of the increasingly more efficient algorithms of the time. Notable examples are the test functions developed by Kursawe [31] and Viennet [32]. Some test functions are based on analytical engineering optimisation problems, such as those introduced by Deb et al. [33] and Poloni et al. [34]. Later, it was realised that the numerical performance ought to be scrutinised using more diverse and challenging sets of test problems. This led to the development of test suites, such as ZDT [35] and DTLZ [19], which later culminated in yearly black-box optimisation benchmark challenges [36]. Another field of test function development concentrates on multi-modal problems [28]. Considering the vast amount of test functions available in literature, only a small selection can be examined throughout this paper. Hence, the functions featured in the benchmark section of

this paper are chosen to highlight and discuss specific features of the algorithm. Results on additional test functions are given in Appendix B.

To compare the benchmark results of multi-objective optimisers, several methods have been applied in the literature. These range from a visual inspection of the points on the Pareto frontier [33] to a variety of performance metrics, which can be grouped into cardinality, accuracy and diversity metrics [37]. Usually, objective functions in engineering optimisation are numerically very expensive, so the focus is on cardinality-based performance assessment. To this end, the yield ratio metric is proposed, which measures the number of Pareto-optimal solutions in relation to the total number of samples generated in the optimisation run. Furthermore, the hypervolume metric is employed, which can be regarded as a combined accuracy and diversity metric [38, 39, 40].

Ultimately, the most important aspect of verifying an engineering optimisation algorithm is the actual application to real engineering problems. To this end, two structural dynamics problems are solved using the proposed method as well as the benchmark algorithm NSGA2. The first problem considers the identification of the dynamic transfer function of a floor slab in a timber building. The second problem demonstrates the use of multi-objective optimisation in vibration-based damage localisation of a steel girder mast via finite element model updating.

2. Single-objective Global Pattern Search algorithm

This paper presents a multi-objective extension of the deterministic Global Pattern Search (GPS) algorithm. The concept and numerical properties of GPS are summarised in this section. A more detailed description of the single-objective algorithm can be found in a previously published article [29]. For this purpose, the bounded, unconstrained, non-linear and derivative-free optimisation problem is defined

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

where f is the objective function and $\boldsymbol{x}$ is comprised of n design variables. Further, the design variable space is defined by the volume of a hypercube

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

where $\boldsymbol{x}_{\mathrm{lb}}$ and $\boldsymbol{x}_{\mathrm{ub}}$ are the lower and upper bounding vectors, respectively.

The basis of the Global Pattern Search algorithm is the classical Pattern Search approach [41]. As their names suggest, a search pattern is employed by this class of algorithms in order to generate new sampling locations. This pattern is created by performing a one-at-a-time permutation around a base vector $\boldsymbol{b}$ using a step width vector $\boldsymbol{w}$. The pattern generation scheme and the resulting sampling locations $\boldsymbol{s}_j$ are depicted in Figure 1.

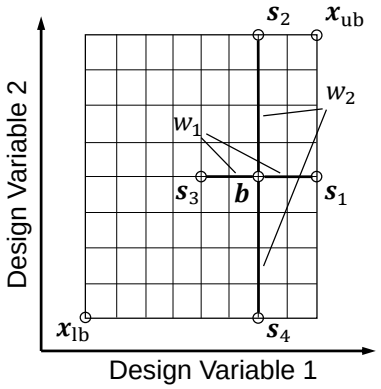


Figure 1: Sampling pattern of the pattern search approach in two dimensions.

To produce a global optimiser using this scheme, more than one base vector is employed in each iteration. These base vectors are determined in each iteration by sorting all previously sampled points according to their objective values and selecting the points with the lowest objective values. The number of points selected as base vectors in each iteration is the only parameter of the GPS algorithm and is denoted as T. The parameter T therefore determines the number of simultaneously tracked globally best points. This major change with regard to Pattern Search yields an effective non-linear global optimisation algorithm. T can be interpreted much like the population size in an evolutionary algorithm, since it controls the diversity of the search pattern. For a setting of $T = 1$ the algorithm tracks only the best point. This means that it behaves like the local Pattern Search algorithm and converges locally. Thus, using $T = 1$, only convex objective functions can be reliably optimised. Higher values of T lead to increasingly more global exploration of the design space. Hence, for $T > 1$, problems with multiple local minima can be solved. The minimum value of T, which leads to global convergence for a specific problem, is dependent on the topology of the specific objective function.

The GPS algorithm is initialised by using the centre of the design space as the first sampling point. The step width vector $\boldsymbol{w}$ is initially set to half the size of the design variable domain. Therefore, the samples subsequently obtained using the axis-aligned search pattern are located on the borders of the design space. During the further course of the optimisation run, the step width is successively halved. Hence, the search pattern implicitly defines a grid, which all samples are situated on [42]. Through reduction of the step width, this grid is iteratively refined which leads to increasingly finer resolutions. Exactly halving the step width also implies that the subdivided grid is compatible with all previous grids. This means that every other point of the subdivided grid coincides with the previous grid. The step width is controlled by halving it when no novel samples can be generated by the search pattern. This functionality is enabled by employing a sample cache, which memorises past sampling points. The sample cache also increases the efficiency of the algorithm by discarding sampling points, which have been visited in previous iterations.

Mathematically, the search pattern is defined as positive and negative variations along the axes of the design variable space

$$s_{j,i} = b_i + \delta_{ij} w_i, \qquad s_{n+j,i} = b_i - \delta_{ij} w_i, \tag{3}$$

where the indices i and j denote the design variable and sample number, respectively. The Kronecker function δ_{ij} is used to denote the one-at-a-time variation. This scheme results in the generation of two samples per design variable for each base vector, as illustrated in Figure 1. The sampling pattern of the GPS algorithm thus resembles a grid, which results from the axis-aligned search pattern.

A numerical example using the Himmelblau test function [43] is shown in Figure 2. The definition of this test function, as well as all other test functions considered in this paper, is given in Appendix A. The Himmelblau test function is a multi-modal function with four equal-valued global minima. The objective value space of this function is illustrated in Figure 2 using contour lines. The samples of the algorithm are colour-coded according to the iteration in which they were generated. The initial sample is situated in the centre of the design variable space and is therefore indicated with a dark blue colour. The samples from the last iterations of this optimisation run are indicated with red colour and are located near the global optima of the Himmelblau test function. This analytical example also showcases the regular sampling grid and its refinement close to the optima of the test function. Another property demonstrated in this example is that the GPS algorithm can be used to reliably solve multi-modal optimisation problems, i.e. problems with several equal-valued minima.

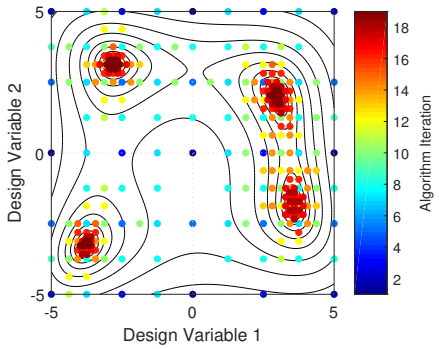


Figure 2: Iteration history of the Global Pattern Search algorithm on the Himmelblau test function using $T = 20$.

A major advantage of the GPS approach is that only one parameter controls the convergence behaviour. The parameter T defines the number of tracked points and thereby controls the balance between convergence rate and design variable space exploration. Having only a single parameter means that GPS can be tuned to new optimisation tasks without much numerical experimentation. Furthermore, it prevents the emergence of complicated interactions between parameters.

Another benefit stems from the deterministic nature of the algorithm: Since the grid is independent of

the parameter choice, samples of previous runs can be reused to speed up the optimisation process of later runs. This strongly reduces the numerical effort when an increase of the parameter T is deemed necessary in a rerun.

3. Multi-objective Global Pattern Search algorithm

In this paper, the design principle of the GPS algorithm is extended to account for multi-objective problems. This extension is designed to retain the benefits of the grid-based deterministic approach as well as having only a single parameter. The following subsections introduce parts of the algorithm in more detail and point out the extensions with respect to the single-objective algorithm. Finally, the complete pseudo-code is given.

3.1. Discretisation of the design variable space

Following the ideas of the single-objective Global Pattern Search algorithm [29], the design variable space is discretised by introducing integer coordinates. The resulting grid is defined using a resolution of 2^N, where N is only limited by the numerical capabilities of the programming language the algorithm is implemented in. For this paper, it is set to $N = 24$. New sampling points are generated in every iteration of the algorithm based on the coordinates of the T globally best points, as described in Section 2. The sample coordinates $\boldsymbol{s}$ are generated on the discrete integer grid and transformed onto the continuous design variable space by

$$x_i = x_{\mathrm{lb},i} + s_i(x_{\mathrm{ub},i} - x_{\mathrm{lb},i}) \cdot 2^{-N}. \tag{4}$$

When the objective function is evaluated for a design variable space coordinate $\boldsymbol{x}$, the result $\boldsymbol{y} = f(\boldsymbol{x})$ is stored to a cache, which prevents redundant evaluations of the objective function in subsequent iterations of the algorithm.

3.2. Pareto-optimality

When designing engineering structures, improving the characteristics of one property often results in degradation of other important properties. By trying to optimise such structures for a specific application, conflicting optimisation goals arise inevitably. A prominent example for conflicting goals are cost and lifetime in design optimisation, where the cheapest design is usually prone to early failure. The most durable design however may be prohibitively expensive. The motivation for multi-objective optimisation is to find multiple solutions to such problems, where each solution represents a trade-off between the objectives. In case of design optimisation, a compromise solution can be obtained, which is durable enough while also being reasonably priced. Multi-objective optimisation can be used whenever conflicting goals arise to numerically obtain trade-off solutions. [44].

Mathematically, the bounded, unconstrained, non-linear and derivative-free multi-objective optimisation problem arises

$$\text{minimise } \boldsymbol{f}(\boldsymbol{x}) \qquad \text{for } \boldsymbol{f} \in \mathbb{R}^m,\ \boldsymbol{x} \in \mathbb{R}^n, \tag{5}$$

where $\boldsymbol{f}$ is the m-dimensional objective function and $\boldsymbol{x}$ comprises n design variables. A sketch of the objective value space is shown in Figure 3.

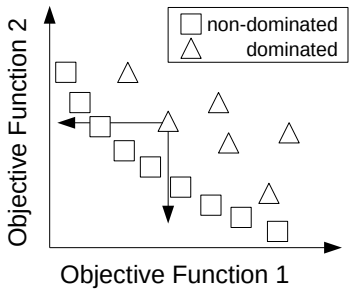


Figure 3: Sketch of objective value space in multi-objective optimisation and Pareto dominance.

In this work, we follow the usual definitions for Pareto dominance, i.e. a point $\hat{\boldsymbol{x}}$ is on the Pareto frontier, if there exists no point $\boldsymbol{x}$ such that $\boldsymbol{f}(\boldsymbol{x}) \preceq \boldsymbol{f}(\hat{\boldsymbol{x}})$ [44]. In Figure 3, the points marked by squares constitute the non-dominated set. The points marked as dominated are Pareto-dominated by several other points and are therefore not on the Pareto frontier. Simply put, for a two-objective problem this means that there must be no point below or to the left of a point on the frontier, as illustrated by the arrows in Figure 3.

3.3. Extension to multi-objective optimisation

The process of identifying the non-dominated set in a set of points is referred to as non-dominated sorting [44]. This involves the numerical sorting of the objective function values and the application of the Pareto dominance operator [45]. Due to the high computational cost involved, the optimal algorithmic implementation is still an active topic of research [46]. Non-dominated sorting is not only used to identify the final non-dominated set resulting from an optimisation run, but is also applied as a part of optimisation algorithms themselves. For instance, the Non-dominated Sorting Genetic Algorithm NSGA2 [3] derives its name from this feature. NSGA2 also includes the notion of secondary frontiers, which are obtained by repetitive application of non-dominated sorting. Figure 4 illustrates this concept and shows how a second and third frontier emerge in parallel to the primary frontier. Numerically, the second frontier is obtained by subtracting the non-dominated set from the whole set of points and applying non-dominated sorting to the remaining subset. This process can be repeated, providing i-th frontiers. These frontiers can contain arbitrary numbers of samples, depending on their coordinates in the objective value space.

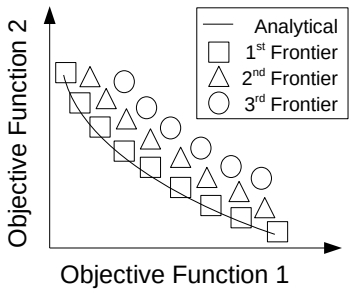


Figure 4: Illustration of secondary frontiers resulting from repeated non-dominated sorting.

The secondary frontiers are employed in the sample generation scheme whenever the first frontier does not contain enough samples to account for the minimum number required by the algorithm. For instance, NSGA2 requires twice as many samples as its population size for its sample generation [2]. As evolutionary algorithms like NSGA2 are designed to maintain a constant population size, the case may arise where not all points on an i-th frontier can be included in the determination of the next sample generation. Thus, a mechanism of reducing the number of points on a frontier has to be employed. In case of NSGA2, this is achieved by using a crowding distance metric. However, taking only a subset of points into account may lead to a statistical bias and thus to a non-uniform coverage of the design variable space. To circumvent this issue, the proposed deterministic multi-objective GPS algorithm takes the frontiers into account as a whole. Due to the varying numbers of samples on the i-th frontiers, this leads to a varying number of points being tracked in each iteration of the algorithm. Hence, the proposed deterministic multi-objective algorithm does not simultaneously track a constant number of globally best solutions. Instead, it tracks at least T globally best solutions from the objective function space. This set of tracked samples from the objective space are referred to as 'hall of fame' in the following.

The updating scheme for the 'hall of fame' of the proposed MOGPS is shown schematically in Figure 5. In this plot, the set of points sampled in all previous iterations is referred to as $\boldsymbol{Y}$, with $|\boldsymbol{Y}|$ indicating the number of samples contained in it. This set is split using repetitive non-dominated sorting to yield the primary and several secondary frontiers, as indicated by division lines in Figure 5. Hence, the cumulative number of samples in the i-th frontiers is $|\boldsymbol{Y}|$. The parameter T is then used to decide how many frontiers are used to populate the 'hall of fame' $\boldsymbol{h}$. In the example shown in Figure 5, the first, second and third frontiers are included in the 'hall of fame', since the first and second frontiers alone would include less than T samples.

The design of the 'hall of fame' updating scheme comes with the benefit of not needing an algorithm for the reduction of points on a frontier, as is the case for evolutionary algorithms like NSGA2. Further, the variable number of tracked globally best solutions represents a contrast to the single-objective GPS algorithm. Thereby, the most important consequence of the proposed scheme for tracking non-dominated solutions is

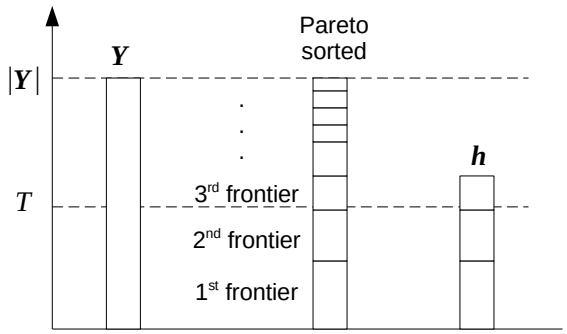


Figure 5: Illustration of the 'hall of fame' updating scheme based on non-dominated sorting and the parameter T.

that the whole primary frontier is always tracked. In late iterations of an optimisation run, this can lead to several thousand points being tracked simultaneously. When compared to tracking only a constant number of globally best solutions, as is the case for evolutionary algorithms including NSGA2, this has a positive effect: Since especially samples near the primary frontier are used as the base vectors in every iteration, virtually all new samples will eventually be situated on the frontier. This leads to a very high percentage of non-dominated samples in late iterations of the MOGPS algorithm, which is demonstrated analysis of the yield ratio metric in Section 4.

There is however a downside inherent to grid-based multi-objective approaches in general: For highly non-linear problems, the grid-based sampling in the design variable space can lead to a highly non-uniform resolution of the resulting non-dominated set, since the non-linearity distorts the sampling grid.

3.4. Details of the algorithm

The procedure for the 'hall of fame' update is shown as a pseudo-code in Algorithm 1. In the proposed algorithm, the updated 'hall of fame' indices $\hat{\boldsymbol{h}}$ are initialised with an empty set and then iteratively filled using the matrix of previously sampled points $\boldsymbol{Y}$. In this iterative update, a set $\boldsymbol{e}$ is computed which excludes the indices to all points already belonging to $\hat{\boldsymbol{h}}$. Using the set $\boldsymbol{e}$, non-dominated sorting is applied and the resulting frontier is added to $\hat{\boldsymbol{h}}$. This process is repeated until at least T indices are stored in $\hat{\boldsymbol{h}}$ or $\hat{\boldsymbol{h}}$ includes all previously sampled points contained in $\boldsymbol{Y}$. The latter case represents the stop criterion for the first few iterations of the algorithm.

Algorithm 1 Update the 'hall of fame'

$\hat{\boldsymbol{h}} \leftarrow \emptyset$
while $|\hat{\boldsymbol{h}}| < T \wedge |\hat{\boldsymbol{h}}| < |\boldsymbol{Y}|$ **do**
  $\boldsymbol{e} \leftarrow \{j : j \in [1..|\boldsymbol{Y}|], j \notin \hat{\boldsymbol{h}}\}$ {Exclude hall of fame}
  $\hat{\boldsymbol{h}} \leftarrow \hat{\boldsymbol{h}} \cup \text{nonDominatedSort}(\boldsymbol{Y}_{\boldsymbol{e}})$ {Add non-dominated set to hall of fame.}
end while

The multi-objective Global Pattern Search scheme is shown in Algorithm 2. As in the single-objective GPS, the caching scheme works with integer coordinates while the evaluation of the objective function takes place in $\mathbb{R}^n$. The 'hall of fame' $\boldsymbol{h}$ comprises indices of the sample cache instead of coordinates to facilitate the set operations needed for the 'hall of fame' update.

Algorithm 2 Multi-objective Global Pattern Search

$w_i \leftarrow 2^{N-1}$ {Initialise step width vector}
$\boldsymbol{S} \leftarrow (\boldsymbol{w})$, $\boldsymbol{h} \leftarrow (1)$ {Initialise sample cache and 'hall of fame'}
loop
  $\boldsymbol{b}_k \leftarrow \boldsymbol{S}(h_k)$ {Get base coordinates using 'hall of fame'}
  for $k = 1$ **to** $|\boldsymbol{h}|$ **do**
    Generate $2n$ sampling coordinates $\boldsymbol{s}_j$ for $\boldsymbol{b}_k$
    Clamp sampling coordinates $\boldsymbol{s}_j$ to design variable space $[0, 2^N]$
    Deduplicate $\boldsymbol{s}_j$ using cache, calculate $\boldsymbol{x}_j$ from $\boldsymbol{s}_j$
    $\boldsymbol{y}_j \leftarrow f(\boldsymbol{x}_j)$ {Sampling}
    Update cache $\boldsymbol{S}$ and objective value matrix $\boldsymbol{Y}$
  end for
  $\hat{\boldsymbol{h}} \leftarrow$ updateHallOfFame()
  if $|\hat{\boldsymbol{h}}| \neq |\boldsymbol{h}|$ **then**
    $\boldsymbol{h} \leftarrow \hat{\boldsymbol{h}}$
    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 using truncating division}
end loop

During run-time, the algorithm needs to store the coordinates of every sampled point as well as the corresponding objective function values. This is facilitated by the matrices $\boldsymbol{S}$, containing the integer coordinates $\boldsymbol{s}_j$, and $\boldsymbol{Y}$, containing the objective function vectors $\boldsymbol{y}_j$, respectively. The sample coordinates are needed to deduplicate samples, while the objective function values are needed for non-dominated sorting.

When using the MOGPS algorithm on a single-objective problem, non-dominated sorting becomes equivalent to linear sorting and the 'hall of fame' update thus becomes identical to that of the single-objective GPS algorithm. This means that the single-objective algorithm is a subset of the multi-objective method. Therefore, the proposed MOGPS can be regarded as a multi-objective generalisation of GPS. Further, since the

number of samples generated in each iteration is even larger than in the single-objective case, the algorithm is ideally suited for parallel computing.

4. Benchmarks

In this section, the MOGPS algorithm is numerically compared to the benchmark algorithm NSGA2 by means of analytical optimisation test functions. The objective of this section is to verify that the numerical performance of the proposed MOGPS method is appropriate for practical engineering optimisation tasks. The focus is on discussing various characteristics of the considered algorithms on the basis of different kinds of optimisation problems rather than proving superior numerical properties. NSGA2 is chosen as the benchmark algorithm because it is arguably the most successful derivative-free multi-objective optimiser to date. Despite its age of 20 years, NSGA2 is still among the most efficient methods [16].

In the following, the performance and convergence properties of the proposed optimisation algorithm are discussed using three test functions: the Poloni [34], the Kursawe [31] and the Two-on-one [47] test function. Due to the prevalence of meta-heuristic approaches, most test problems are not designed with grid-based approaches in mind. Since grid-based approaches generate sample points on the boundaries and along the axes of the design variable space in early stages of the optimisation run, they are very likely to converge on optima in these regions. In essence, this means that grid-based approaches have an unfair advantage when applied to optimisation problems with optimal regions on the boundaries or axes. The analytical optimisation test functions featured in this section were chosen for detailed analysis, because the optimal solutions of these problems are mostly situated neither directly on boundaries nor aligned to the axes. This helps to reduce possible bias effects introduced by the test functions themselves when considering the grid-based MOGPS algorithm. The popular ZDT benchmark problems by Zitzler et al. [35] have axis-aligned optimal regions on the boundaries and are therefore deliberately not discussed in detail due to said bias issue. Nonetheless, results for ZDT as well as additional test problems are reported in Appendix B.

The scope of this benchmark is additionally limited to unconstrained optimisation, since constraint handling involves aspects beyond the scope of the optimisation approach itself. The restriction to unconstrained problems thus enables a 'fair' comparison between algorithms, because the performance on constrained problems strongly depends on the settings and type of the employed constraint handling technique. However, a prototype version of the MOGPS algorithm has successfully been applied to a practical constrained engineering problem by Berger et al. [48]. Further, the engineering example described in Section 5.2 also incorporates an inequality constraint.

4.1. Quality metrics

The results of the benchmark runs are assessed on the basis of numerical quality metrics. These metrics are chosen for the comparison of the numerical performance with respect to the total number of samples as

well as the volume of the dominated objective function space. This approach is motivated by engineering optimisations with numerically expensive objective functions like finite element calculations. In such optimisation problems, the computing time is proportional to the number of calculated samples and optimisers thus have to be assessed accordingly. To this end, two quality metrics are employed: the hypervolume metric, which measures the dominated objective function space and the yield ratio, which measures the ratio of non-dominated samples generated by the algorithm.

The hypervolume metric is used extensively in literature [39, 40, 37] and evaluates the dominated hypervolume relative to a fixed reference point

$$\mathrm{HV} \equiv \Lambda \left(\bigcup_{\boldsymbol{y} \in \boldsymbol{Y}} \{ \boldsymbol{y}' \mid \boldsymbol{y} \prec \boldsymbol{y}' \prec \boldsymbol{y}_{\mathrm{ref}} \} \right), \tag{6}$$

where Λ denotes the Lebesgue measure, $\boldsymbol{Y}$ is the set of evaluated objective function vectors and $\boldsymbol{y}_{\mathrm{ref}}$ is a reference point, which has to be dominated by all points on the Pareto frontier. Equation 6 essentially describes the space enclosed between the non-dominated points generated by the algorithm and the fixed reference point. Because the reference point has to be dominated by all points, a higher hypervolume metric indicates a better approximation of the Pareto frontier. The maximum value depends on the geometry of the Pareto frontier associated with the optimisation problem as well as on the chosen reference point. Since the hypervolume metric is based on the dominated space, it is influenced by the Pareto-optimality of individual points as well as the resolution of the frontier.

Further, the yield ratio is proposed as a measure of the efficiency in finding non-dominated points

$$\mathrm{YR} \equiv \frac{\text{number of non-dominated samples}}{\text{number of total samples}} = \frac{|\{\hat{\boldsymbol{y}} \in \boldsymbol{Y} \mid \nexists\, \boldsymbol{y} \in \boldsymbol{Y} : \boldsymbol{y} \preceq \hat{\boldsymbol{y}}\}|}{|\boldsymbol{Y}|}, \tag{7}$$

where $|\boldsymbol{Y}|$ denotes the cardinality of the set $\boldsymbol{Y}$. This concept is similar to other cardinality-based metrics, including the comparative metrics Ratio of Non-dominated Individuals [37, 49, 50] as well as the Set Coverage Metric [35]. A similarly motivated intrinsic metric is the Overall Non-dominated Vector Generation Ratio [51]. Yield ratios are defined in the range $[0, 1]$, where low values indicate poor performance in terms of finding non-dominated points. Contrarily, the value 1 resembles an ideal solution to the optimisation problem, in which every sample is also a non-dominated solution. Thus, a high yield ratio indicates that the algorithm efficiently uses samples to improve the resolution of the non-dominated set. A theoretical perfect algorithm would therefore maximise the hypervolume metric while achieving a yield ratio of 1.

4.2. Poloni test function

The Figures 6 and 7 show results obtained with MOGPS for the Poloni test function. The definition of this function is given in Appendix A. The algorithm is parameterised using $T = 16$ and terminated after 500 objective function evaluations. The figures display all samples generated in the optimisation run as well as

the final non-dominated set frontier. The frontier is highlighted using a colour gradient based on the first objective value to provide a mapping between the objective and the design variable space.

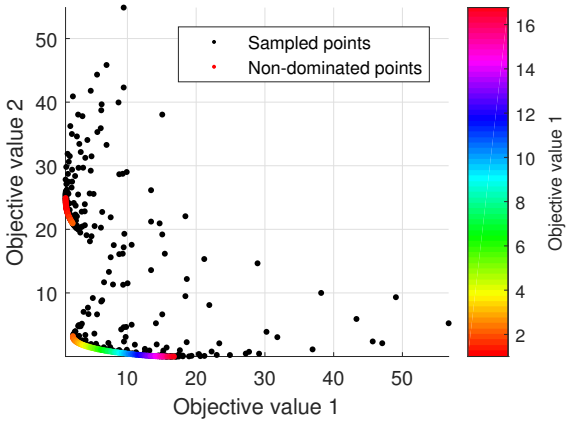


Figure 6: Objective value space obtained using MOGPS, showing all samples (black dots) and the final non-dominated set using a colour bar with respect to objective value 1.

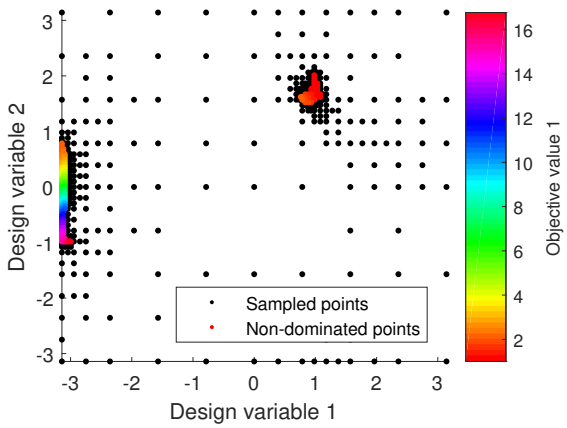


Figure 7: Design variable space and sampling pattern for the Poloni test function obtained using MOGPS. Colour bar same as Figure 6.

The sampling pattern displayed in Figure 7 reflects the grid-based and self-similar search pattern of the proposed algorithm. The plot demonstrates a good balance between global sampling and a strong convergence near the Pareto-optimal areas. Regarding the objective value space shown in Figure 6, the concentration of sampling points near the Pareto frontier is evident. This behaviour enables the achievement of a high yield ratio.

In order to compare the performance and convergence properties of the MOGPS algorithm to NSGA2, the dependencies of the yield ratio and the hypervolume metric on the number of objective function evaluations are evaluated for different parameter sets for the value T and the population size, respectively. The results are shown in Figure 8. For this comparison, the benchmark algorithm NSGA2 is run with the mutation probability $p_{\text{mut}} = n^{-1}$ and the crossover probability $p_{\text{cross}} = 0.9$, as suggested by the original authors [3]. The simulated binary crossover distribution parameters are set to $\eta_c = 10$ and $\eta_m = 10$. To account for the metaheuristic nature of NSGA2, the results are averaged over 10 optimisation runs.

Figure 8 a) shows the trend of the hypervolume indicator over the number of objective function evaluations for both algorithms and different settings. The hypervolume indicator is plotted using logarithmic scaling and is normalised using the maximum hypervolume achieved in all runs. The horizontal trends of the hypervolume metric for $T = 1$ and for pop $= 8$ indicate local convergence. This occurs because parts of the frontier are not discovered and the algorithms thus concentrate on only one part of the frontier. Higher settings for T and pop lead to a higher global coverage of the design variable space, which may result in the recovery of more separate parts of the frontier, but at the same time slows down the convergence. The hypervolume metric was

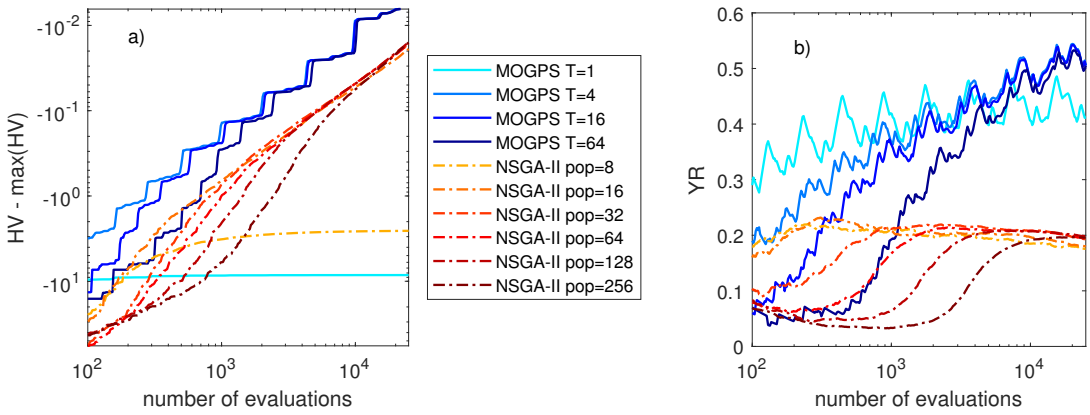


Figure 8: Evolution of a) the hypervolume metric and b) the yield ratio over the number of objective function evaluations on the Poloni test function for different parameter sets for MOGPS and NSGA2.

evaluated with the reference point (20|30) in the objective value space, leading to max(HV) $\approx$ 536.09. This maximum value corresponds to the origin of the ordinate of Figure 8 a). Except for the two aforementioned parameter settings, both algorithms continually converge towards the Pareto frontier, which is evident from the continually increasing hypervolume metric.

Figure 8 b) shows the evolution of the yield ratio metric over the course of the optimisation runs. The cases $T = 1$ and pop $= 8$ exhibit a high yield ratio metric, however they have been identified as locally convergent using Figure 8 a) and can therefore be disregarded. In general, the proposed MOGPS algorithm is able to consistently improve the yield ratio over the course of an optimisation run. Low settings of T lead to high yield ratios early in the optimisation run. As the algorithm becomes increasingly more local for lower T, this behaviour is expected. The reason for the convergence of the yield ratios of the runs with $T > 1$ is that the number of non-dominated samples increases for high numbers of objective function evaluations. As the 'hall of fame' is always populated with the non-dominated samples, the included globally best solutions become identical for high numbers of objective function evaluations, as discussed in Section 3.3. In contrast, the NSGA2 runs exhibit peaks in the yield ratio, where the corresponding number of objective function evaluations depend on the chosen population sizes: Lower population sizes are suitable for optimisation runs with few objective function evaluations, whereas higher settings lead to a more global coverage of the design variable space. The yield ratios of NSGA2 eventually converge once the whole population is located in Pareto-optimal areas of the design variable space.

Concerning the spread of non-dominated samples, MOGPS by design cannot provide uniform distances in the objective value space. An example for this behaviour is shown in Figure 9. This plot focuses on the lower part of the Pareto frontier shown in Figure 6. The Pareto frontier is shown in a cyan colour and the non-dominated points obtained with a run of the MOGPS algorithms are shown in magenta. In this plot, the non-dominated samples are positioned on a stepped pattern. This behaviour is the result of the grid-based

approach, where the samples are uniformly distributed in the design variable space. Due to the non-linear nature of the Poloni test function, the grid-like sampling of the design variable space is distorted when it is transformed to the objective value space. This results in non-uniform distributions, which also vary along the Pareto frontier.

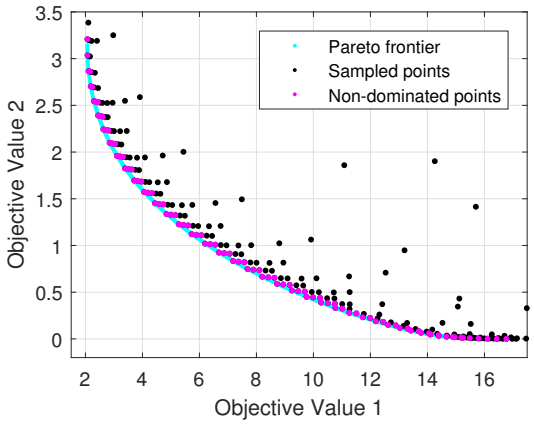


Figure 9: Part of the objective value space of the Poloni test function showing the distribution of samples obtained with MOGPS using $T = 16$ and 500 evaluations.

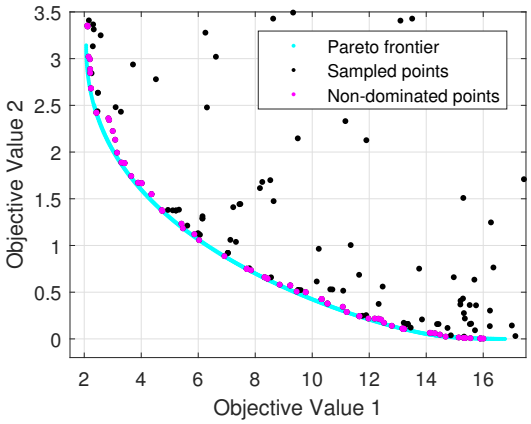


Figure 10: Part of the objective value space of the Poloni test function showing the distribution of samples obtained with NSGA2 using pop $= 32$ and 500 evaluations.

A practical mitigation of the non-uniform coverage issue is not easily achievable for MOGPS, since a regularisation of the design variable space would be necessary, which requires prior knowledge of the underlying problem. Deb [44] also discussed the biasing problems associated with non-uniform objective functions. His solution to this issue was the design of an explicit crowding distance metric for NSGA2. This results in an unbiased and undistorted distribution of the non-dominated points, as depicted in Figure 10.

Due to the positioning of the Pareto optimal areas, the Poloni test function is ideally suited to additionally discuss the issue of local convergence. The trend of the hypervolume metric shown in Figure 8 indicates local convergence for $T = 1$. The same behaviour is observed when NSGA2 is run with the insufficiently small population size pop $= 8$. Hence, a setting of $T = 1$ is used to investigate the local convergence phenomenon for MOGPS. Figure 11 displays the resulting objective value space of the Poloni function. In this case, the algorithm fails to discover the upper part of the Pareto frontier.

This convergence behaviour is caused by a failure to track sampling points close to the position of the upper section of the Pareto frontier in the first few iterations. Figure 12 shows the corresponding design variable space. Due to the initially small size of the 'hall of fame', the algorithm converges rapidly without much global exploration. This in turn means that the algorithm entirely misses the part of the Pareto frontier close to $\boldsymbol{x} = [1\ 2]^T$ and thus converges locally. In contrast, by choosing a sufficiently high value for T, global convergence is achieved, as demonstrated in Figure 7. As is the case for the population size parameter in NSGA, some numerical experimentation or experience is required in order to determine the minimum setting

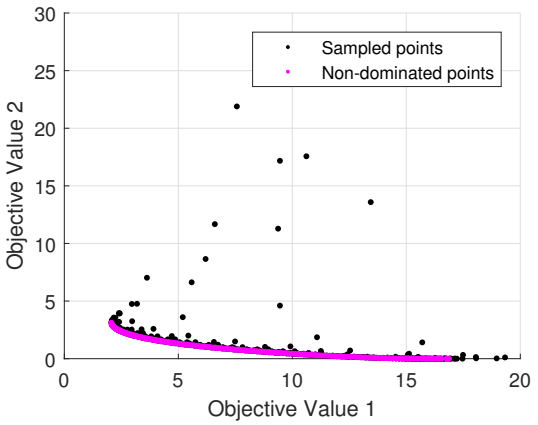


Figure 11: Objective value space of the Poloni test function obtained with MOGPS using $T = 1$ and 500 evaluations.

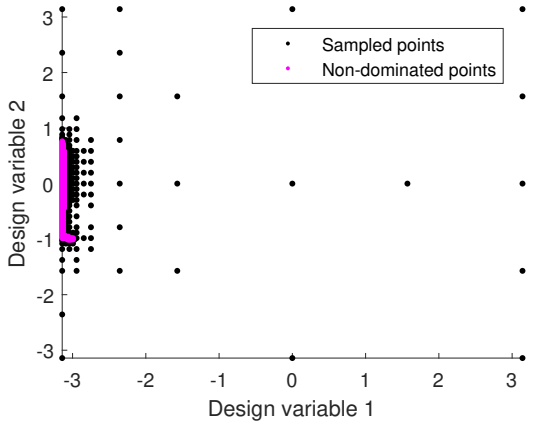


Figure 12: Design variable space of the Poloni test function obtained with MOGPS using $T = 1$ and 500 evaluations.

for T, which leads to global convergence.

4.3. Kursawe test function

The performance is additionally examined on the three-variable optimisation problem by Kursawe [31], as defined in Appendix A. Figure 13 displays the non-dominated set obtained using the proposed MOGPS algorithm. The frontier is discovered very well, whereby 3000 objective function evaluations and $T = 16$ were considered. In the corresponding design variable space, depicted in Figure 14, it is noticeable that the optimal lines, areas and volumes, which belong to the parts of the Pareto frontier, are rendered with high fidelity.

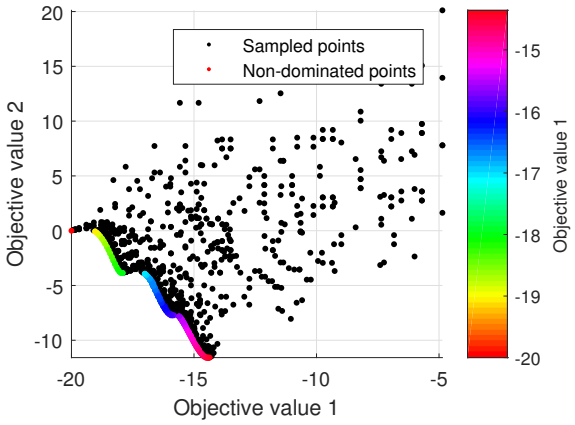


Figure 13: Objective value space, showing all samples (black dots) and the final non-dominated set (colour bar w.r.t objective 1) obtained using MOGPS for the Kursawe test function.

Again, the results are compared to those obtained with the implementation of NSGA2. This comparison

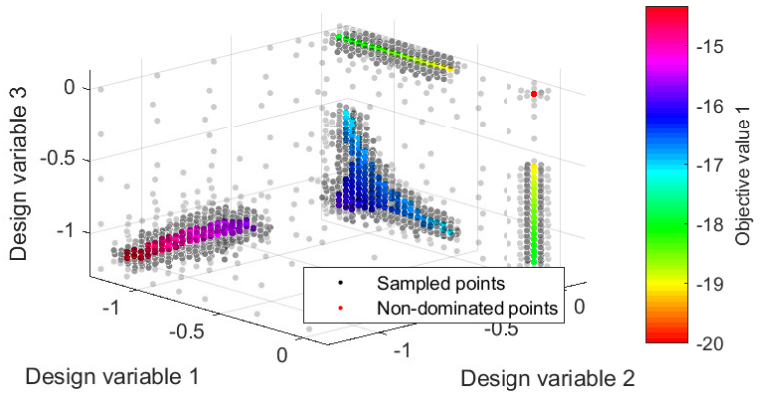


Figure 14: Design space and sampling pattern obtained using MOGPS for the Kursawe test function. Colour bar same as Figure 13.

is illustrated in Figure 15. The hypervolume metric is evaluated with the reference point $(-15|5)$ in the objective value space, leading to max(HV) ≈ 44.72. Again, the ordinate of 15 a) is normalised using the maximum hypervolume metric achieved by all runs to allow for a logarithmic scale plot. In case of the Kursawe test function, a setting of $T = 1$ does not lead to local convergence and thus provides the highest numerical performance in both metrics. The number of objective function evaluations is indicated using a logarithmic scale on the abscissa of Figures 15 a) and b). The MOGPS approach performs similar to NSGA2 for low numbers of objective function evaluations and becomes more efficient in both performance metrics after approximately 3000 evaluations. Regarding the yield ratio metric, MOGPS performs significantly better as the number of evaluations increases.

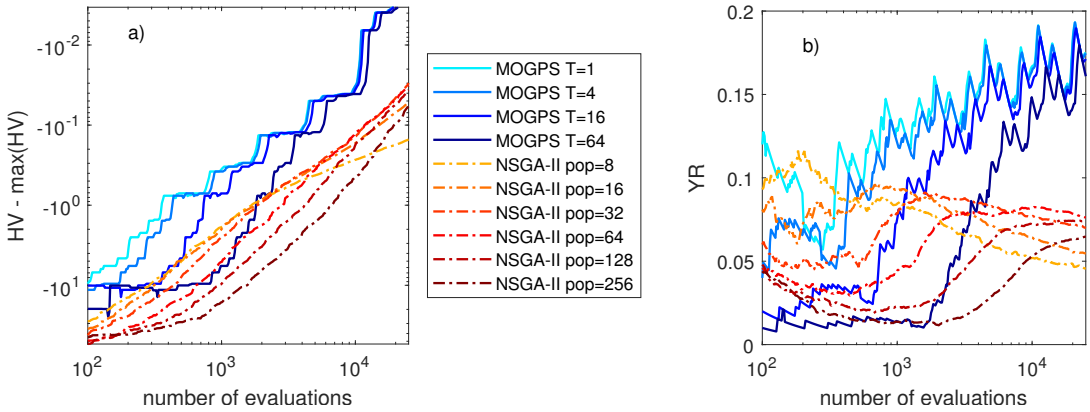


Figure 15: Evolution of a) the hypervolume metric and b) the yield ratio over the number of objective function evaluations on the Kursawe test function for different parameter sets.

The Kursawe test function is used to demonstrate the added complexity associated with additional parameters in meta-heuristic optimisation algorithms like the considered NSGA2 algorithm. A population size

of 64 is used, as it leads to a good performance in terms of the hypervolume metric as well as the yield ratio metric, as shown in Figure 15. For the purpose of this demonstration, a parametric study is conducted for one of the five parameters of the NSGA2 algorithm: The crossover probability p_{cross} is varied using increments of 0.1. Figure 16 shows the evolution of the performance metrics over the number of objective function evaluations. The hypervolume metric shows that the best convergence is achieved when choosing $p_{\text{cross}} = 1$. The yield ratio shows that low values of p_{cross} lead to higher percentages of non-dominated points. Therefore, the value of 0.9 chosen for the preceding optimisation runs with NSGA2 is indeed an acceptable compromise solution.

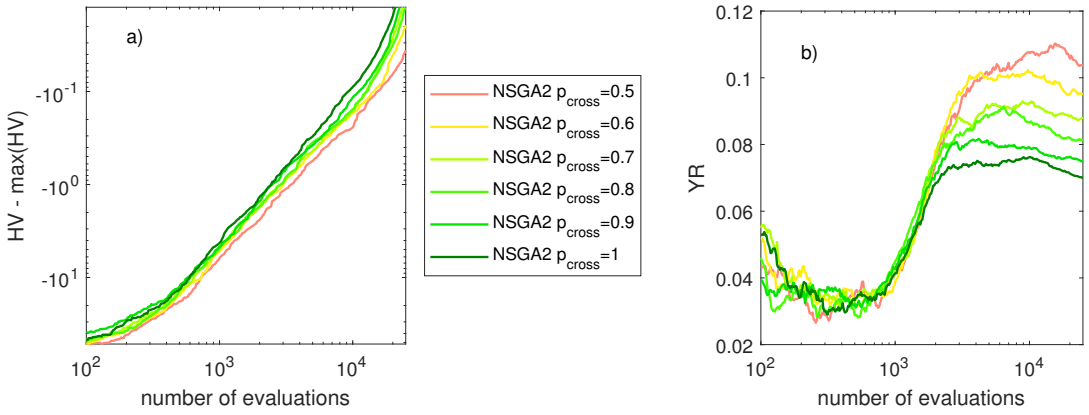


Figure 16: Evolution of a) the hypervolume metric and b) the yield ratio of NSGA2 over the number of objective function evaluations on the Kursawe test function for different settings of the crossover probability.

However, the effect of parameter settings on the algorithms performance is strongly dependent on the optimisation problem. Further, the sensitivity of each individual parameter also depends on the settings of every other parameter. For novel optimisation problems with no prior knowledge of the solutions, the performance will thus always be sub-optimal. This issue is amplified further, the more parameters an algorithm has. Hence, optimisation algorithms intended for numerically expensive engineering problems should be designed in a way that a minimal number of algorithm parameters need to be tuned.

The proposed MOGPS method has only the parameter T which controls the optimisation process. This means that just this parameter needs to be tuned to the optimisation problem at hand and no consideration has to be given to possible interactions with other parameters. Thus, it is especially well suited for computationally demanding engineering optimisation problems like finite element model updating, where numerical experimentation with parameter settings would be very time-consuming.

4.4. Two-on-one test function

The Two-on-one test function was designed by Preuss et al. [47] to test the behaviour of multi-objective optimisation algorithms on multi-modal problems and is defined in Appendix A. As in single-objective op-

timisation, multi-modal multi-objective problems have two or more regions in the design space which map onto the same objective function values. Preuss et al. came to the conlusion that the tested algorithms were not capable of properly handling multi-modal problems. Recently, evolutionary multi-objective algorithms were retrofitted with special subroutines in order to tackle these issues [28]. However, these come at the cost of additional algorithmic complexity and are beyond the scope of this paper.

For the sake of consistency, NSGA2 is used on this analytic test function even though it is not designed to handle multi-modal problems. This serves to showcase the difficulties that most unmodified multi-objective optimisers encounter when solving problems of this kind. The samples in the design variable space resulting from runs of NSGA2 and MOGPS are shown in Figures 17 and 18, respectively.

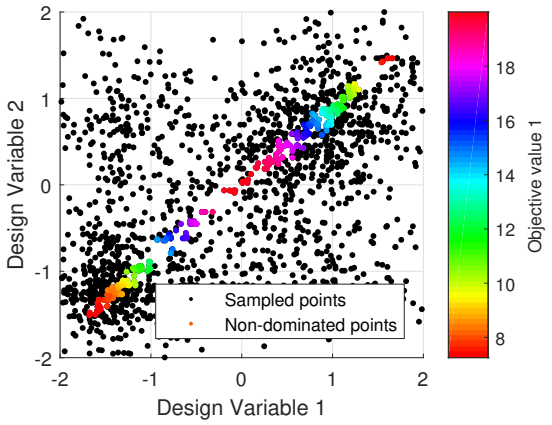


Figure 17: Design space and sampling pattern for the Two-on-one test function obtained using NSGA2.

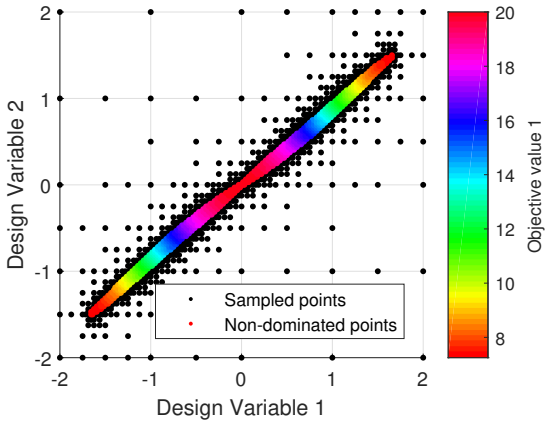


Figure 18: Design space and sampling pattern for the Two-on-one test function obtained using MOGPS.

In both instances, the algorithms are given a limit of 2000 objective function evaluations. MOGPS is run with $T = 16$, while NSGA2 is run with a population size of 64, a crossover probability of 0.9 and a mutation probability of 0.5. As evident from the plots, NSGA2 has a sub-optimal convergence behaviour and hits Pareto optimal areas only sporadically. In contrast, The MOGPS algorithm exhibits a clear convergence towards the optimal areas and produces a uniform coverage. This multi-modal capability is inherited from the single-objective GPS method, as explained in Section 2. The ability to solve multi-modal problems without modifications to the algorithm is however related to the specific implementation of the Pareto sorting algorithm. Measures need to be in place to make sure that duplicate points in the non-dominated set are not deduplicated. Otherwise, convergence only occurs on one of the multi-modal frontiers since samples related to the other ones are disregarded.

5. Engineering examples

Benchmark test functions and numerical comparisons to other algorithms are a valuable tools to verify optimisation methods. But, after all, optimisation algorithms are applied in practice to solve non-linear

engineering problems. In this section, two practical problems from the field of structural dynamics are solved using both the proposed MOGPS method and NSGA2. The first example is about the identification of transfer functions for a floor slab in a timber building. The second example demonstrates the usage of multi-objective optimisation in vibration-based damage localisation using finite element analysis.

In addition to the rather brief practical examples reported in this paper, a prototype version of the MOGPS algorithm has been employed for more advanced optimisation tasks. Berger et al. [48] used it for the design optimisation of aero turbine repairs and Haldar et al. [52] applied it to the design optimisation of a bistable laminate actuator.

5.1. Transfer function identification for a floor slab

In the structural dynamics of buildings, the identification of transfer functions is an important task, since vibration amplitudes can become critical when resonant excitation occurs. In this example, a slab of a building constructed from cross-laminated timber is considered, which was subjected to dynamic testing as depicted in Figure 19. The vibration measurements of one of the building floors is carried out using a modal hammer, which enables recording the hammer force, as shown in Figure 20. The vibration response of the building floor is measured using geophones. The impulse response displayed in Figure 21 suggests, that the floor vibration is dominated by two frequencies: A low frequency component with a low modal damping as well as a higher frequency component, which quickly decays. Physically, these two frequencies can be associated to the first bending mode of the floor and the eigenmode of the screed.

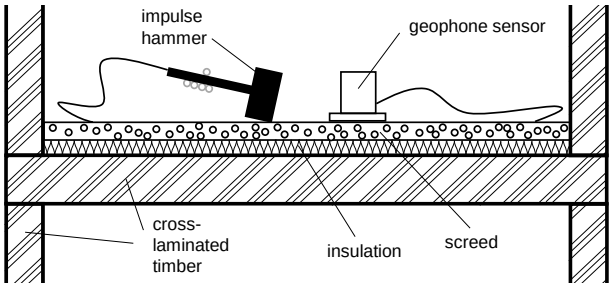


Figure 19: Sketch of the design of the timber slab and the dynamic experimental setup.

A two-objective optimisation is applied to identify the eigenfrequency, the damping and the amplification of the two modes. The objectives are the Root Mean Square (RMS) error of the time series and the peak amplitude error. On the one hand, a low RMS error indicates a good fit of amplitudes, eigenfrequencies and damping. The accurate prediction of the peak amplitude on the other hand is important for vibration prognosis, since this value is frequently used as the basis of comfort assessments. Because transfer functions with a low RMS error tend to underestimate the peak amplitude and vice versa, the two objectives are conflicting and thus usually cannot be satisfied at once.

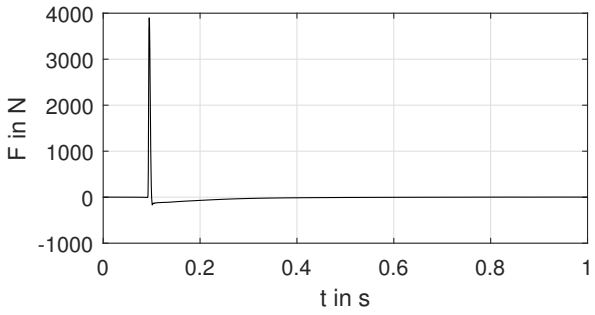


Figure 20: Time series of hammer force measurement.

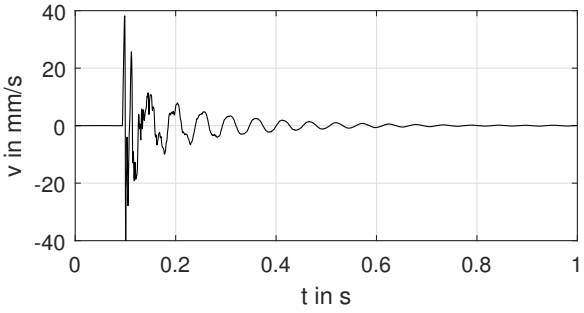


Figure 21: Time series of measured floor response.

Using discrete time filters, a mechanical system consisting of two damped harmonic oscillators is modelled. The oscillators are both excited by the input signal and the sum of their outputs is used as the prediction of the vibration velocity. The harmonic oscillators are described using the complex poles

$$s_{1k,2k} = 2\pi f_{0k}\left(-D_k \pm j\sqrt{1-D_k^2}\right), \tag{8}$$

where k denotes the index of the oscillator and f_{0k} and D_k are the eigenfrequencies and critical damping fractions, respectively. This model is discretised using time increments of T by applying a matched z-transformation to obtain the poles of the discrete time model

$$z_{1k} = e^{s_{1k}T},\ z_{2k} = e^{s_{2k}T} \tag{9}$$

The z-transformed poles can be inserted into the discrete time transfer functions

$$G_k(z) = \frac{1-z}{(z-z_{1k})(z-z_{2k})}, \tag{10}$$

$$G(z) = \beta\ (\alpha G_1(z) + (1-\alpha)G_2(z)), \tag{11}$$

where $G_k(z)$ denotes the transfer function of the individual oscillators with force as the input and velocity as the output. $G(z)$ is the combined transfer function with both oscillators connected in parallel, where β is

Table 1: Design variables for the transfer function and their permissible range.

Design variable	Unit	Minimum	Maximum
β	-	3	30
α	-	0	1
f_{01}	Hz	10	30
f_{02}	Hz	40	80
D_1	%	0.5	10
D_2	%	5	50

the total amplitude factor and α is a weighting factor for the individual harmonic oscillators. This discrete transfer function is applied to the hammer force time series using a Direct-Form-II infinite impulse response filter implementation [53]

$$v(z) = G(z)\ F(z), \tag{12}$$

where $F(z)$ denotes the hammer force and $v(z)$ denotes the floor velocity calculated by the model. The filter output time series $v[i]$ is compared to the measured floor velocity signal $v_{\text{meas}}[i]$. The two objectives RMS and peak error can be expressed as

$$\boldsymbol{f} = \begin{pmatrix} \epsilon_{\text{RMS}} \\ \epsilon_{\text{peak}} \end{pmatrix} = \begin{pmatrix} \sqrt{\dfrac{1}{N}\sum_{i}^{N}(v[i] - v_{\text{meas}}[i])^2} \\ \left|\max_{i}^{N}(|v[i]|) - \max_{i}^{N}(|v_{\text{meas}}[i]|)\right| \end{pmatrix}, \tag{13}$$

where N denotes the number of samples contained in the measured time series. The design variable vector is defined as

$$\boldsymbol{x} = \Big[\log(\beta)\ \alpha\ \log(f_{01})\ \log(f_{02})\ \log(D_1)\ \log(D_2)\Big]^T, \tag{14}$$

where most of the variables are logarithmised as a means of regularisation of the design variable space. The ranges of the variables are stated in Table 1.

The optimisation is run with 10 000 objective function evaluations using both MOGPS and NSGA2. The parameters are set to $T = 16$ for MOGPS and to a population size of 64 for NSGA2. The Pareto frontier resulting from these runs are compared in Figure 22. Taking all samples on the Pareto frontier into account, the results for the eigenfrequency and damping are used to visualise the identified system parameters, as shown in Figure 23. Both algorithms retrieve a high-resolution Pareto frontier and identify the modal parameters in similar ranges. The results of MOGPS exhibit less stochastic variation in the objective space when compared to NSGA2, which is due to its deterministic nature.

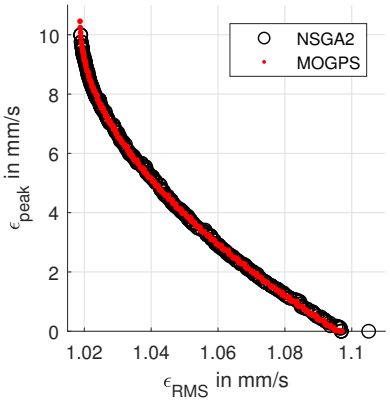


Figure 22: Non-dominated points resulting from dynamic parameter identification from floor slab measurement using NSGA2 and MOGPS.

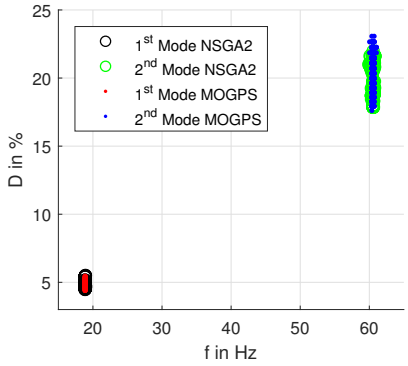


Figure 23: Identified eigenfrequencies and damping ratios for the first and second eigenmodes.

The spread in Figure 23 can be interpreted as uncertainty in the identification, where the first eigenmode is identified with higher certainty than the second one. While the scatter of the frequencies is low, the uncertainty of the damping ratio is significant. This finding coincides with an uncertainty quantification that was carried out based on the Bayesian method developed by Siu-Kui Au [54] using the same measurement data. However, a detailed analysis of the uncertainty is beyond the scope of this work.

5.2. Damage localisation on a girder mast structure

Structural health monitoring is another field where the usage of multi-objective optimisation can be advantageous. In this example, the MOGPS algorithm is applied to the finite element model updating of a girder mast structure. The mast is 9 m tall, made of construction steel and situated in an outdoor test facility illustrated in Figure 24. The structure is dynamically excited by environmental conditions and the structural acceleration is measured using 18 piezoelectric accelerometers situated on the nine measurement levels shown in Figure 25.

A structural damage was introduced by cutting the braces in the second bay of the structure, as also indicated in Figure 25. The modal parameters of the structure are identified by applying the frequency domain decomposition method [55] on 10-minute data sets. One of the data sets was recorded before the damage event and the other one thereafter. The data sets are chosen so that the vibration amplitudes as well as the material temperature are roughly the same. This way, data contamination due to differing environmental conditions is prevented. The identified modes shapes and eigenfrequencies are shown in Figure 26 for the undamaged and the damaged data sets, respectively.

The model updating is performed using a finite element model of the girder mast structure, in which the legs and braces are modelled using first-order beam elements. Materials and cross-sections are assigned according to the as-measured dimensions of the structure. The modal analysis of the model is carried out

Figure 24: Photograph of girder mast structure.

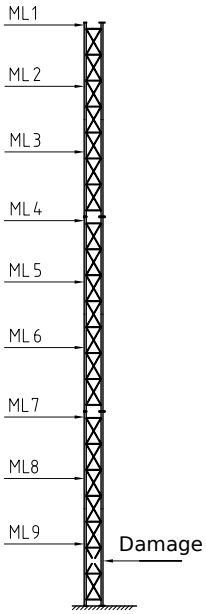


Figure 25: Sketch of structure and damage location.

using the finite element solver Abaqus. The resulting mode shapes for the undamaged structure resemble the measured mode shapes, as shown in in Figure 27.

A damage distribution function is employed to control the structural stiffness of the model [56, 57]. The optimisation problem is set up to modify the distribution function and match the simulation results to the measured structural behaviour. This damage distribution function is a Gaussian distribution and parameterised using three parameters

$$\boldsymbol{x} = \begin{bmatrix} \mu & D & \sigma \end{bmatrix}^T, \tag{15}$$

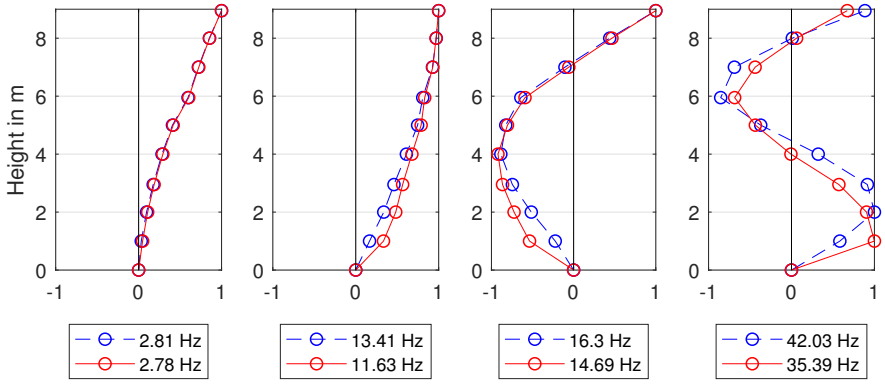
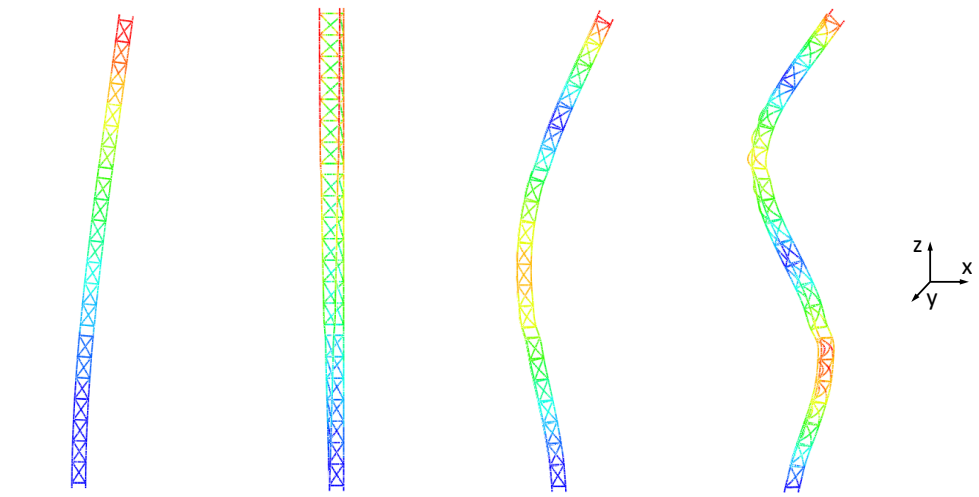


Figure 26: Mode shapes and eigenfrequencies identified from measurement data for the structure shown in Figures 24 and 25. Undamaged state indicated using dashed blue lines, damaged state indicated using solid red lines.



1st bending	1st torsion	2nd bending	3rd bending
3.28 Hz	15.48 Hz	19.0 Hz	48.6 Hz

Figure 27: Finite element results for mode shapes and eigenfrequencies of the undamaged state.

where μ is the centre of the damage, D is the damage intensity and σ describes the standard deviation of the damage distribution. These parameters are then used to prescribe the stiffness scaling factors θ_i for each bay of the structure

$$\theta_i(\mathbf{x}) = 1 - \left(D \cdot F\left(r_i | \mu, \sigma, 0\,\mathrm{m}, 9\,\mathrm{m}\right) - \sum_{j=1}^{i-1} 1 - \theta_j(\mathbf{x}) \right), \tag{16}$$

where i denotes the index of the bay, r_i is the bay's height above ground and F is a truncated Gaussian cumulative distribution function. The stiffness scaling factors θ_i are used to scale the elastic modulus of the respective bays prior to the modal analysis of the model. Even though the actual damage was introduced only to the braces, the stiffness scaling is applied to the legs as well in each bay of the model. This is done to obtain a more realistic scenario, where it is not known in advance if the damage occurred in a brace or in a leg. By applying the stiffness scaling to the whole bay, the model updating method can therefore be made sensitive to damages both in the braces and the legs.

The vibration modes taken into account in the model updating procedure are chosen to achieve a high damage sensitivity. As Figure 26 indicates, the second and third bending modes as well as the first torsion mode change significantly due to the damage. In contrast, the first bending mode exhibits a very low damage sensitivity and is thus not used in the updating. To utilise the full information contained in the modal parameters, the mode shapes as well as the eigenfrequencies are used for the updating. This is facilitated by computing the difference between the modal parameters of the reference state and the damaged state. The

differences in the considered eigenfrequencies and mode shapes are summed up to yield two updating errors

$$\begin{pmatrix} \epsilon_f \\ \epsilon_m \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{1}{N_{\text{modes}}} \sum_{k=1}^{N_{\text{modes}}} \left(\frac{f_{\text{S1},k}(\boldsymbol{x}) - f_{\text{S0},k}}{f_{\text{S0},k}} - \frac{f_{\text{M1},k} - f_{\text{M0},k}}{f_{\text{M0},k}} \right)^2} \\ \sqrt{\frac{1}{N_{\text{modes}}} \sum_{k=1}^{N_{\text{modes}}} \left| (\boldsymbol{m}_{\text{S1},k}(\boldsymbol{x}) - \boldsymbol{m}_{\text{S0},k}) - (\boldsymbol{m}_{\text{M1},k} - \boldsymbol{m}_{\text{M0},k}) \right|^2} \end{pmatrix}, \tag{17}$$

where ϵ_f is the eigenfrequency error and ϵ_m is the mode shape error. The eigenfrequencies f and mode shape vectors $\boldsymbol{m}$ are denoted with a subscript $(\cdot)_{\text{S}}$ for simulated and $(\cdot)_{\text{M}}$ for measured data. The subscript $(\cdot)_0$ refers to the undamaged state, while $(\cdot)_1$ refers to the damaged state. The design variables only influence the simulation results for the damaged case, while all other terms of Equation 17 remain constant during the optimisation run. The mode shapes used to compute ϵ_m are normalised, similar to the approach of the enhanced COMAC metric [58].

In previous works on model updating, the mode shape and eigenfrequency error metrics are frequently weighted and summed up to yield a compound metric [59]. However, the weighting factors which yield the best model updating result are unknown prior to the optimisation run. The error metrics can also be weighted equally [60], but this assumption may lead to sub-optimal identification results. The application of multi-objective optimisation can remove this shortcoming [61], since both error metrics can be solved for simultaneously. In fact, the Pareto frontier resulting from the multi-objective formulation contains the optimal solutions for all conceivable weighting factor combinations. The multi-objective approach thus gives additional information about the uncertainty of the localisation and thus represents a contrast to a single-objective optimisation where only a single solution results from the optimisation without any additional information.

Since the value range of the stiffness scaling factors is not restricted to positive values by Equation 16, it is possible that negative θ_i values may arise for low values of σ. However, negative stiffness values would lead to meaningless finite element results. To avoid this issue, all models with negative stiffness values are rejected prior to finite element calculation. Since this approach creates a discontinuity in the objective function, a constraint is added in order to soften this discontinuity. Therefore, the minimum stiffness scaling factor is used to formulate an inequality constraint, which acts to restrict values below 15% of the original stiffness. This leads to the formulation of the bounded and constrained multi-objective optimisation problem

$$\begin{aligned} &\text{minimise} \begin{pmatrix} \epsilon_f \\ \epsilon_m \end{pmatrix} \\ &\text{s.t.} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 9 & 0.3 & 2 \end{bmatrix}^T \\ &\text{s.t.} \min_i (\theta_i) > 0.15. \end{aligned} \tag{18}$$

The constraint is enforced using the exterior linear penalty method [62]. The resulting penalty term is applied equally to both objective function values.

The MOGPS and NSGA2 algorithms are run with 2000 objective function evaluations using $T = 16$ and a population size of 64, respectively. The resulting non-dominated solutions are shown in Figure 28. In this plot, the NSGA2 results are shown as black circles, while the MOGPS results are colour-coded according to ϵ_f. Both algorithms converge to similar solutions, with MOGPS achieving a higher resolution since it is able to find more non-dominated points.

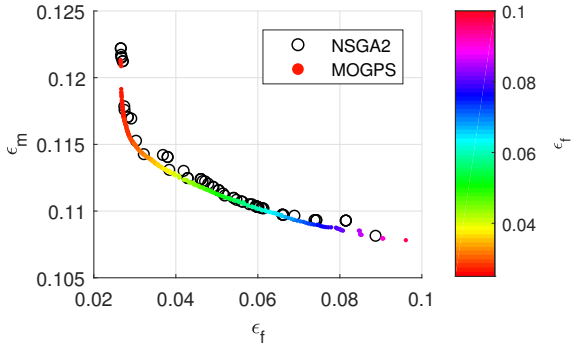


Figure 28: Pareto frontier of the multi-objective model updating problem.

The non-dominated points identified by the algorithms can also be illustrated by the stiffness scaling factors associated with the corresponding design variables. Figure 29 shows the distributions with a colour coding based on the eigenfrequency error ϵ_f. In this figure, the distributions belonging to the lowest ϵ_f (i.e. red colour) indicate a damage at the root of the structure. The distributions with the highest ϵ_f (i.e. purple colour) point to a damage at a height of approximately 1 m. The most accurate damage localisation is produced by solutions belonging to $\epsilon_f \approx 0.04$ (i.e. yellow colour), which place the centre of the distribution in the bay where the damage actually occurred as indicated in Figure 29 by the dashed lines.

The use of multi-objective optimisation in this example enables a comprehensible insight into the most probable damage distributions of the structure. The assessment of the stiffness distribution functions in Figure 29 shows that no matter how the eigenfrequency and mode shape errors are weighted, the damage is always identified close to the base of the structure. This means that the multi-objective model updating approach achieves a high confidence in terms of damage localisation. Further, the MOGPS algorithm exhibits favourable performance in this practical example as it is able to provide a high-resolution Pareto frontier. The NSGA2 algorithm is also able to converge to the Pareto frontier but cannot achieve the fine resolution of the MOGPS algorithm.

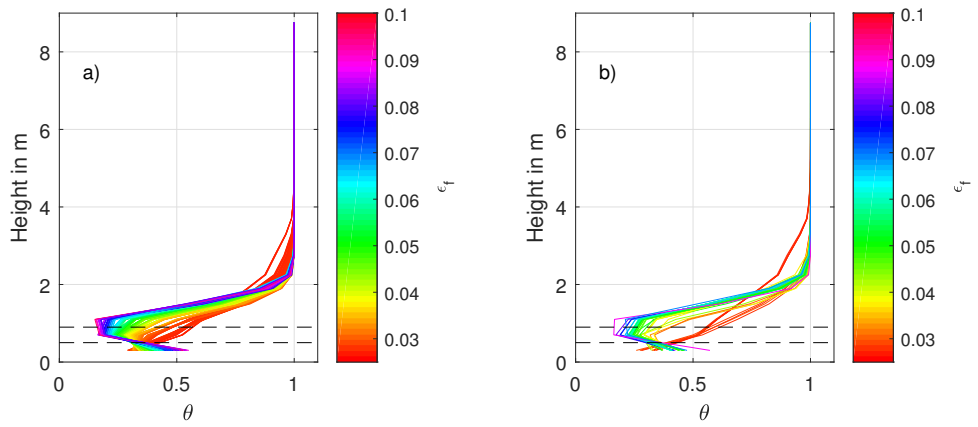


Figure 29: Stiffness scaling factors computed for the Pareto frontier. MOGPS results shown in panel a) and NSGA2 results shown in panel b). Position of experimentally damaged bay indicated with dashed lines.

6. Summary and outlook

This paper presented a novel deterministic multi-objective optimisation algorithm based on the Global Pattern Search approach. The algorithm lends itself to practical engineering optimisation problems due to its numerical performance and because it has only one control parameter. The approach was evaluated using analytical test functions as well as practical engineering examples.

Considering the three analytical test functions discussed in section 4, the efficiency of the proposed multi-objective extension of the Global Pattern Search approach is conclusively demonstrated. An in-depth comparison of the performance of the proposed method to the benchmark algorithm NSGA2 was conducted. It is shown in detail, that the proposed MOGPS algorithm performs similar to NSGA2 and, depending on the circumstances, even outperforms it. Additional results on more analytical test problems are reported in the supplementary data, which support and supplement the results discussed in Section 4.

The numerical results for the yield ratio metric indicate that the MOGPS algorithm is able to increase the ratio of non-dominated points with respect to the number of objective function evaluations. The proposed algorithm should thus be preferred, when a high resolution of Pareto-optimal solutions is sought. For very low numbers of evaluations, random-number based algorithms such as NSGA2 are more appropriate, because stochastic sampling generates more non-dominated solutions early in the optimisation run.

As shown in section 5, MOGPS performs well when applied to engineering problems. Two examples are discussed, in which the practical utility of the proposed method is examined and compared to NSGA2. In both examples, NSGA2 produced satisfactory results, however MOGPS consistently achieved a higher quality, which was reflected in the high-resolution Pareto frontiers obtained. This is expected to be the case for most practical applications, since the single control parameter of MOGPS makes it easier to achieve good results when compared to other algorithms which require problem-specific tuning of multiple interdependent

parameters.

A topic not covered in this work is the usage of alternative sampling patterns. The grids proposed by Belitz and Bewley [42] may be used to further increase the efficiency when solving high-dimensional problems. Future research will focus on the benefits and downsides associated with the incorporation of such patterns.

Taking into account the results obtained in this work, we conclude that the seldom used deterministic approaches for multi-objective optimisation tasks are a viable alternative for engineering problems.

7. Acknowledgments

We greatly acknowledge the financial support of Deutsche Bundesstiftung Umwelt (research project *Gebrauchstauglichkeit und Komfort von dynamisch beanspruchten Holztragwerken im urbanen mehrgeschossigen Hochbau*, AZ 34548/01-25) that enabled this work.

References

[1] A. Custódio, M. Emmerich, J. Madeira, Recent developments in derivative-free multiobjective optimization, Computational Technology Reviews 5 (1) (2012) 1–31. doi:10.4203/ctr.5.

[2] K. Deb, Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction, Springer London, London, 2011, Ch. 1, pp. 3–34. doi:10.1007/978-0-85729-652-8_1.

[3] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: Parallel Problem Solving from Nature PPSN VI, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 849–858. doi:10.1007/3-540-45356-3_83.

[4] C. A. Coello, G. T. Pulido, Multiobjective optimization using a micro-genetic algorithm, in: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, GECCO'01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 274–282.

[5] J. Horn, N. Nafpliotis, D. E. Goldberg, A niched pareto genetic algorithm for multiobjective optimization, in: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Vol. 1, 1994, pp. 82–87. doi:10.1109/ICEC.1994.350037.

[6] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms — a comparative case study, in: Parallel Problem Solving from Nature — PPSN V, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 292–301. doi:10.1007/BFb0056872.

[7] A. Konak, D. W. Coit, A. E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliability Engineering & System Safety 91 (9) (2006) 992 – 1007, special Issue - Genetic Algorithms and Reliability. doi:10.1016/j.ress.2005.11.018.

[8] C. A. C. Coello, G. T. Pulido, M. S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 256–279. doi:10.1109/TEVC.2004.826067.

[9] X. Hu, R. Eberhart, Multiobjective optimization using dynamic neighborhood particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02, Vol. 2, 2002, pp. 1677–1681. doi:10.1109/CEC.2002.1004494.

[10] S. Mostaghim, J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), in: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03, 2003, pp. 26–33. doi:10.1109/SIS.2003.1202243.

[11] N. Beume, B. Naujoks, M. Emmerich, Sms-emoa: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181 (3) (2007) 1653 – 1669. doi:10.1016/j.ejor.2006.08.008.

[12] J. Knowles, D. Corne, The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, Vol. 1, 1999, pp. 98–105. doi:10.1109/CEC.1999.781913.

[13] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, TIK-report 103 (2001) 1–21. doi:10.3929/ethz-a-004284029.

[14] T. Voß, N. Hansen, C. Igel, Improved step size adaptation for the mo-cma-es, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO '10, ACM, New York, NY, USA, 2010, pp. 487–494. doi:10.1145/1830483.1830573.

[15] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731. doi:10.1109/TEVC.2007.892759.

[16] D. Brockhoff, T.-D. Tran, N. Hansen, Benchmarking numerical multiobjective optimizers revisited, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, ACM, 2015, pp. 639–646. doi:10.1145/2739480.2754777.

[17] H. Ishibuchi, R. Imada, Y. Setoguchi, Y. Nojima, Performance comparison of nsga-ii and nsga-iii on various many-objective test problems, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3045–3052. doi:10.1109/CEC.2016.7744174.

[18] R. Berger, M. Bruns, A. Ehrmann, A. Haldar, J. HÃďfele, B. Hofmeister, C. HÃijbler, R. Rolfes, Engio âĂŞ object-oriented framework for engineering optimization, Advances in Engineering Software 153 (2021) 102959. doi:10.1016/j.advengsoft.2020.102959.

[19] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, in: Evolutionary multiobjective optimization, Springer, 2005, pp. 105–145. doi:10.1007/1-84628-137-7_6.

[20] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization 11 (4) (1997) 341–359. doi:10.1023/A:1008202821328.

[21] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Transactions on Evolutionary Computation 15 (1) (2011) 4–31. doi:10.1109/TEVC.2010.2059031.

[22] K. Sörensen, Metaheuristics – the metaphor exposed, International Transactions in Operational Research 22 (1) (2015) 3–18. doi:10.1111/itor.12001.

[23] R. Rao, Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems, International Journal of Industrial Engineering Computations 11 (1) (2020) 107–130. doi:10.5267/j.ijiec.2019.6.002.

[24] Y. Evtushenko, M. Posypkin, A deterministic algorithm for global multi-objective optimization, Optimization Methods and Software 29 (5) (2014) 1005–1019. doi:10.1080/10556788.2013.854357.

[25] A. Al-Dujaili, S. Suresh, Dividing rectangles attack multi-objective optimization, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 3606–3613. doi:10.1109/CEC.2016.7744246.

[26] A. L. Custódio, J. A. Madeira, A. I. F. Vaz, L. N. Vicente, Direct multisearch for multiobjective optimization, SIAM Journal on Optimization 21 (3) (2011) 1109–1140. doi:10.1137/10079731X.

[27] P. Alotto, G. Capasso, A deterministic multiobjective optimizer, COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering 34 (5) (2015) 1351–1363. doi:10.1108/COMPEL-03-2015-0117.

[28] R. Tanabe, H. Ishibuchi, A review of evolutionary multimodal multiobjective optimization, IEEE Transactions on Evolutionary Computation 24 (1) (2019) 193–200. doi:10.1109/TEVC.2019.2909744.

[29] B. Hofmeister, M. Bruns, R. Rolfes, Finite element model updating using deterministic optimisation: A global pattern search approach, Engineering Structures 195 (2019) 373 – 381. doi:10.1016/j.engstruct.2019.05.047.

[30] J. D. Schaffer, Some experiments in machine learning using vector evaluated genetic algorithms, Ph.D. thesis, Vanderbilt University, Nashville, TN, USA (1984).

[31] F. Kursawe, A variant of evolution strategies for vector optimization, in: International Conference on Parallel Problem Solving from Nature, Springer, 1990, pp. 193–197. doi:10.1007/BFb0029752.

[32] R. Viennet, C. Fonteix, I. Marc, Multicriteria optimization using a genetic algorithm for determining a pareto set, International Journal of Systems Science 27 (2) (1996) 255–260. doi:10.1080/00207729608929211.

[33] K. Deb, A. Pratap, S. Moitra, Mechanical component design for multiple ojectives using elitist non-dominated sorting ga, in: Parallel Problem Solving from Nature PPSN VI, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 859–868. doi:10.1007/3-540-45356-3_84.

[34] C. Poloni, A. Giurgevich, L. Onesti, V. Pediroda, Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics, Computer Methods in Applied Mechanics and Engineering 186 (2-4) (2000) 403–420. doi:10.1016/S0045-7825(99)00394-1.

[35] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput. 8 (2) (2000) 173–195. doi:10.1162/106365600568202.

[36] N. Hansen, A. Auger, R. Ros, S. Finck, P. Pošík, Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009, in: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 1689–1696. doi:10.1145/1830761.1830790.

[37] N. Riquelme, C. V. Lücken, B. Baran, Performance metrics in multi-objective optimization, in: 2015 Latin American Computing Conference (CLEI), 2015, pp. 1–11. doi:10.1109/CLEI.2015.7360024.

[38] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms — a comparative case study, in: Parallel Problem Solving from Nature — PPSN V, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 292–301. doi:10.1007/BFb0056872.

[39] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132. doi:10.1109/TEVC.2003.810758.

[40] N. Beume, C. M. Fonseca, M. Lopez-Ibanez, L. Paquete, J. Vahrenhold, On the complexity of computing the hypervolume indicator, IEEE Transactions on Evolutionary Computation 13 (5) (2009) 1075–1082. doi:10.1109/TEVC.2009.2015575.

[41] R. Hooke, T. A. Jeeves, "direct search" solution of numerical and statistical problems, J. ACM 8 (2) (1961) 212–229. doi:10.1145/321062.321069.

[42] P. Belitz, T. Bewley, New horizons in sphere-packing theory, part II: lattice-based derivative-free optimization via global surrogates, Journal of Global Optimization 56 (1) (2013) 61–91. doi:10.1007/s10898-012-9866-7.

[43] D. M. Himmelblau, Applied Nonlinear Programming, McGraw-Hill, 1972.

[44] D. Kalyanmoy, et al., Multi objective optimization using evolutionary algorithms, John Wiley and Sons, 2001.

[45] H. T. Kung, F. Luccio, F. P. Preparata, On finding the maxima of a set of vectors, J. ACM 22 (4) (1975) 469–476. doi:10.1145/321906.321910.

[46] P. C. Roy, K. Deb, M. M. Islam, An efficient nondominated sorting algorithm for large number of fronts, IEEE Transactions on Cybernetics 49 (3) (2019) 859–869. doi:10.1109/TCYB.2017.2789158.

[47] M. Preuss, B. Naujoks, G. Rudolph, Pareto set and emoa behavior for simple multimodal multiobjective functions, in: Parallel Problem Solving from Nature-PPSN IX, Springer, 2006, pp. 513–522. doi:10.1007/11844297_52.

[48] R. Berger, B. Hofmeister, C. G. Gebhardt, R. Rolfes, A two-objective design optimisation approach for blending repairs of damaged compressor blisks, Aerospace Science and Technology 105 (2020) 106022. doi:10.1016/j.ast.2020.106022.

[49] M. Kim, T. Hiroyasu, M. Miki, S. Watanabe, SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 742–751. doi:10.1007/978-3-540-30217-9_75.

[50] K. C. Tan, T. H. Lee, E. F. Khor, Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons, Artificial intelligence review 17 (4) (2002) 251–290. doi:10.1023/A:1015516501242.

[51] D. A. Van Veldhuizen, Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations, Ph.D. thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, USA (1999).

[52] A. Haldar, E. Jansen, B. Hofmeister, M. Bruns, R. Rolfes, Analysis of novel morphing trailing edge flap actuated by multistable laminates, AIAA Journal 58 (7) (2020) 3149–3158. doi:10.2514/1.J058870.

[53] J. O. Smith, Introduction to digital filters: with audio applications, Vol. 2, W3K publishing, 2007.

[54] S.-K. Au, Uncertainty law in ambient modal identification—Part II: Implication and field verification, Mechanical Systems and Signal Processing 48 (1) (2014) 34 – 48. doi:10.1016/j.ymssp.2013.07.017.

[55] R. Brincker, L. Zhang, P. Andersen, Modal identification of output-only systems using frequency domain decomposition, Smart Materials and Structures 10 (3) (2001) 441–445. doi:10.1088/0964-1726/10/3/303.

[56] M. Bruns, B. Hofmeister, C. Hübler, R. Rolfes, Damage Localization Via Model Updating Using a Damage Distribution Function, in: Structural Health Monitoring 2019, DEStech Publications, Inc, Lancaster, PA, 2019, pp. 1–10. doi:10.12783/shm2019/32202.

[57] M. Bruns, B. Hofmeister, T. Grießmann, R. Rolfes, Comparative study of parameterizations for damage localization with finite element model updating, in: Proceedings of the 29th European Safety and Reliability Conference (ESREL), Research Publishing Services, Singapore, 2019, pp. 1125–1132. doi:10.3850/978-981-11-2724-3_0713-cd.

[58] D. L. Hunt, Application of an enhanced coordinate modal assurance criterion, in: 10th International modal analysis conference, Vol. 1, 1992, pp. 66–71.

[59] E. Simoen, G. De Roeck, G. Lombaert, Dealing with uncertainty in model updating for damage assessment: A review, Mechanical Systems and Signal Processing 56-57 (2015) 123–149. doi:10.1016/j.ymssp.2014.11.001.

[60] K. Schröder, C. G. Gebhardt, R. Rolfes, A two-step approach to damage localization at supporting structures of offshore wind turbines, Structural Health Monitoring 17 (5) (2018) 1313–1330. doi:10.1177/1475921717741083.

[61] J. Naranjo-Pérez, M. Infantes, J. Fernando Jiménez-Alonso, A. Sáez, A collaborative machine learning-optimization algorithm to improve the finite element model updating of civil engineering structures, Engineering Structures 225 (2020) 111327. doi:10.1016/j.engstruct.2020.111327.

[62] C. A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer Methods in Applied Mechanics and Engineering 191 (11) (2002) 1245 – 1287. doi:10.1016/S0045-7825(01)00323-1.

Appendix A. Test functions

Appendix A.1. Himmelblau function

$$f(\boldsymbol{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{A.1}$$

$$\text{s.t.} \begin{bmatrix} -5 & -5 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 5 & 5 \end{bmatrix}^T.$$

See Himmelblau [43].

Appendix A.2. Poloni function

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} \left[1 + (A_1 - B_1(x_1, x_2))^2 + (A_2 - B_2(x_1, x_2))^2\right] \\ (x_1 + 3)^2 + (x_2 + 1)^2 \end{pmatrix} \tag{A.2}$$

where

$$\begin{aligned} A_1 &= 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2) \\ A_2 &= 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2) \\ B_1(x_1, x_2) &= 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2) \\ B_2(x_1, x_2) &= 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2) \end{aligned}$$

$$\text{s.t.} \begin{bmatrix} -\pi & -\pi \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} \pi & \pi \end{bmatrix}^T.$$

See Poloni et al. [34].

Appendix A.3. Kursawe function

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} \sum_{i=1}^{2} \left[-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right] \\ \sum_{i=1}^{3} \left[|x_i|^{0.8} + 5\sin\left(x_i^3\right)\right] \end{pmatrix} \tag{A.3}$$

$$\text{s.t.} \begin{bmatrix} -5 & -5 & -5 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 5 & 5 & 5 \end{bmatrix}^T.$$

See Kursawe [31].

Appendix A.4. Two-on-one function

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{pmatrix} x_1^4 + x_2^4 - x_1^2 + x_2^2 - cx_1x_2 + dx_1 + 20 \\ (x_1 - k)^2 + (x_2 - l)^2 \end{pmatrix} \tag{A.4}$$

where

$$k = l = d = 0$$

$$c = 10$$

$$\text{s.t.} \begin{bmatrix} -2 & -2 \end{bmatrix}^T \leq \boldsymbol{x} \leq \begin{bmatrix} 2 & 2 \end{bmatrix}^T.$$

See Preuss et al. [47].

Appendix B. Supplementary Material

The supplementary data can be found at `https://doi.org/10.15488/10319`.