DOI 10.2478/ama-2023-0034

ABILITY OF BLACK-BOX OPTIMISATION TO EFFICIENTLY PERFORM SIMULATION STUDIES IN POWER ENGINEERING

Lukas PETERS*, Rüdiger KUTZNER*, Marc SCHÄFER, Lutz HOFMANN*****

*University of Applied Sciences and Arts Hannover, Faculty I – Electrical Engineering and Information Technology, Ricklinger Stadtweg 120, 30459 Hannover, Germany
**Siemens Energy Gas and Power Combustion Systems, Mellinghofer Str. 55, 45473 Mülheim a.d. Ruhr, Germany
***Leibniz University Hannover, Institute of Electric Power Systems, Electric Power Engineering Section, Appelstraße 9a, 30167 Hannover, Germany

lukas.peters@hs-hannover.de, ruediger.kutzner@hs-hannover.de, schaefermarc@siemens-energy.com, hofmann@ifes.uni-hannover.de

received 14 November 2022, revised 9 February 2023, accepted 12 February 2023

Abstract: In this study, the potential of the so-called black-box optimisation (BBO) to increase the efficiency of simulation studies in power engineering is evaluated. Three algorithms ("Multilevel Coordinate Search" (MCS) and "Stable Noisy Optimization by Branch and Fit" (SNOBFIT) by Huyer and Neumaier and "blackbox: A Procedure for Parallel Optimization of Expensive Black-box Functions" (blackbox) by Knysh and Korkolis) are implemented in MATLAB and compared for solving two use cases: the analysis of the maximum rotational speed of a gas turbine after a load rejection and the identification of transfer function parameters by measurements. The first use case has a high computational cost, whereas the second use case is computationally cheap. For each run of the algorithms, the accuracy of the found solution and the number of simulations or function evaluations needed to determine the optimum and the overall runtime are used to identify the potential of the algorithms in comparison to currently used methods. All methods provide solutions for potential optima that are at least 99.8% accurate compared to the reference methods. The number of evaluations of the objective functions differs significantly but cannot be directly compared as only the SNOBFIT algorithm does stop when the found solution does not improve further, whereas the other algorithms use a predefined number of function evaluations. Therefore, SNOBFIT has the shortest runtime for both examples. For computationally expensive simulations, it is shown that parallelisation of the function evaluations (SNOBFIT and blackbox) and quantisation of the input variables (SNOBFIT) are essential for the algorithmic performance. For the gas turbine overspeed analysis, only SNOBFIT can compete with the reference procedure concerning the runtime. Further studies will have to investigate whether the quantisation of input variables can be applied to other algorithms and whether the BBO algorithms can outperform the reference methods for problems with a higher dimensionality.

Key words: black-box optimisation, power plant engineering, derivative-free optimisation, simulation studies in power engineering

1. INTRODUCTION

Many engineering tasks are, at their core, worst- or best-case analyses and thus optimisation problems in a mathematical sense. Furthermore, for many of these analyses, the use of simulation models is necessary, and simulation models can be expensive to compute. In some cases, the inner structure of models is too complex to directly investigate the influence of certain parameters, and the only way to obtain the needed information is the simulation. Calculations showing this behaviour can be characterised as black-boxes [1] (p. 1). In other applications, the term black-box could also be used for real physical experiments, where the correlation of the conditions of a system and the result of the experiment is unknown. Referring to simulations, it is also possible that models from external sources are given as black-box codes to protect intellectual property. For all these cases, optimisation methods that do not need any information about the structure of the function to be optimised are necessary. These methods are called black-box optimisation (BBO).

To avoid any confusion, the nomenclature used in this article is explained at this point: The term BBO is connected to the terms derivative-free optimisation (DFO) and simulation optimisation. Even though the terms emphasise on different aspects, all three might be applicable to most of the methods categorised with one of them. DFO emphasises the absence of information about the derivatives for the optimisation algorithm, whereas simulation optimisation is linked to the optimisation of experiments with simulation models. In the field of the project, the most important aspect is the universality of the methods for different use cases or objective functions. To emphasise the characteristic of not needing further information about the actual optimised function, the term BBO is the most appropriate for this study and will therefore be used.

There exist various reviews in the field of BBO [2–6], and beneath a high number of applications in biology, medicine, logistics or operations [5] (p. 358), these methods have also been used in engineering, e.g., for shape optimisation [7–12], electronic components [13, 14], control design [15] and power engineering [16, 17]. More recently, different methods have been applied in fluid dynamics [31–33], in chemical engineering [34, 35] or to tune parameters of other algorithms [36].

First algorithms that can be used for BBO were developed in the 1960s. One of those algorithms, the Nelder–Mead Simplex

Algorithm [27], is still used because of its straightforward principle that makes it easily applicable in different scenarios. After the first use of trust region methods [28] in 1969, in the 1970s and 1980s, genetic [29] and stochastic [30] algorithms emerged. Due to the increasing computational capabilities from then on, the number of algorithms and applications has increased enormously. Rios and Sahinids [3] have vividly illustrated the progress in their work.

While the performance comparison of optimisation algorithms is mostly based on academic test functions (see, e.g., [3]), the applications listed previously do focus more on the specific problem to solve and less on the evaluation of different algorithms. Using an algorithm for a single problem allows for tuning this algorithm to efficiently solve this problem, which might lead to a worse performance of this algorithm for other applications.

In this project, a variety of BBO methods were considered for different applications in the field of power engineering and evaluated concerning their potential to simplify engineers' daily work. For this simplification, the key is to find optimisation methods that are easy to implement and use as well as reliable. Following this goal, the present study introduces this topic by evaluating different methods mentioned in the corresponding reviews and answer the question whether BBO methods are competitive to manual procedures that are currently applied.

For the models used to develop and analyse gas and steam turbine control systems, models with this characteristic are used, e.g., to investigate damping capabilities of the control system for dynamic events of the electric grid, to define the highest possible rotational speed in case of a load shedding event or to parametrise the controllers.

Traditionally, mathematical optimisation is performed using information about the function's derivatives. Unfortunately, there are functions whose derivatives are very or even prohibitively expensive to obtain – as it is the case for the field mentioned previously. Consequently, the field of BBO has been driven forwards in mathematics over the past decades as computational power has massively increased.

Literature research and first attempts are promising that BBO can be easily applied to different problems and that BBO algorithms could improve the efficiency of simulation studies, in this case in turbine control system engineering.

In this study, different algorithms were applied to a variety of exemplary problems from the aforementioned sphere of action. The study will answer to which extend BBO can simplify the corresponding studies. The study also shows that BBO methods can – considering both of the algorithms and the accuracy of the found solution – efficiently solve optimisation problems in the field of power plant control engineering. Nevertheless, running parallel functions evaluations and limiting the resolution of the input variables are necessary features for an algorithm to be competitive even for computationally cheap objective functions.

2. SIMULATION STUDIES IN POWER PLANT CONTROL ENGINEERING

In 1999, Lu wrote in "Dynamic modelling and simulation of power plant system" [18] that in the previous decades, different models of power plants had been used to compute steady-state operation as well as to predict dynamics in case of events, like failures or incidents, and – of course – for usage in hardware training simulators.

Currently, the so-called digital twins of turbines or even whole power plants mirror all signals to indicate and forecast a plant's behaviour in real time [19]. Simulation studies are inevitable not only in the development of different components, such as turbine blades, but also in testing logics for control systems.

While there exist models in every complexity for different simulation scenarios to serve the most suitable computation of the plant's behaviour, the efficiency of the simulation studies themselves is not always optimal. In many cases, studies are performed manually and/or by computing large grids of input parameters. These studies might be led by an engineer's experience and therefore miss potentially interesting aspects, and additionally, the duration of executing these studies can be rather long.

Semi-automating parts of the studies might significantly decrease the time consumption of the studies. One idea to automate parts of such studies is using optimisation algorithms specially designed to solve problems with computational expensive functions, which will be referred to as BBO. Serving the aim to solve – at best – all possible problems with a single algorithm in practice, the following two use cases are chosen for a first evaluation of different BBO methods. The two use cases differ enormously in their computational costs and complexity and therefore cover a large percentage of the scale of problems that are subjects for this project.

2.1. Use Case 1: gas turbine overspeed simulation

Fig. 1 shows a load shedding event of an unspecified gas turbine power plant.

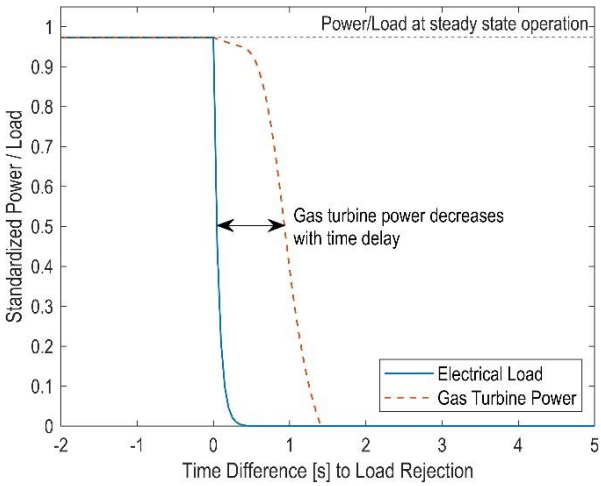


Fig. 1. Electrical load and gas turbine power output during a load shedding event

Here, the load from the electric grid and the effective power generated by the gas turbine are plotted over time. For steady-state operation with a constant rotor speed, both quantities are close to equal, which means that not only the power the gas turbine provides is approximately the same as the power of the electric grid, but also that the torque the gas turbine transmits M_{turb} on the shaft has the same amount as the counter-torque necessary to operate the compressor of the gas turbine M_{comp} and the counter-torque from the electric load M_{load}:

$$M_{turb} + M_{comp} + M_{load} = 0 = J_{shaft} \cdot \dot{\omega} \quad (1)$$

where J_{shaft} is the moment of inertia of the rotating components of the turbine and the generator and $\dot{\omega}$ is the change in the angular velocity of the rotor. Losses resulting, e.g., from friction within the bearings are included within M_{turb}. For any difference in the torques, the shaft will be accelerated or decelerated, as shown in Eq. 1, until the power provided by the gas turbine P_{turb} is reduced or a higher angular velocity is reached at an equilibrium of the torque from the turbine and the counter-torque from the compressor, which depend on the rotational speed:

$$P_{turb} = (M_{turb} + M_{comp}) \cdot \omega \quad (2)$$

Following Eq. 2, it is clear that whenever the load changes, the power of the turbine has to be adjusted to keep the angular velocity ω and the rotational speed $n=\omega/2\pi$ constant.

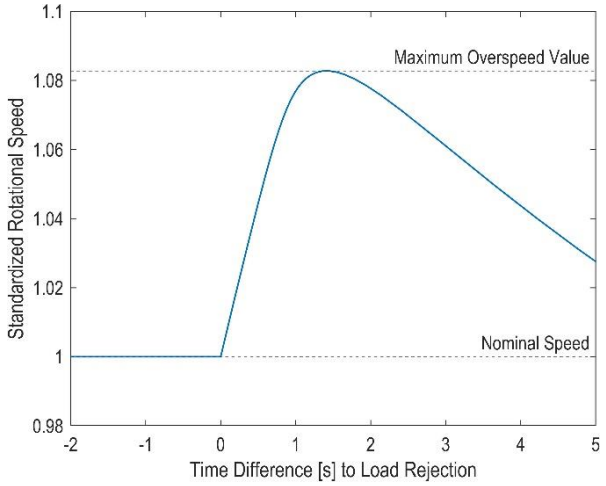


Fig. 2. Rotational speed of a gas turbine during a load shedding event

When the generator is disconnected from the grid, the electrical load does no longer provide a counter-torque, and the power of the turbine will – following Eq. 2 – result in an acceleration of the rotor until the control system has reduced the fuel supply and therefore the power.

A load shedding event (see the blue line in Fig. 1) is a highly dynamic process, and the turbine control systems are not able to immediately adjust the supplied power via the supplied fuel mass flow. Firstly, the fuel gas in the piping volume between the control valve and the combustion chamber (CC) (see Fig. 3) will expand into the CC and therefore be still converted into thermal power, which results in a delay between lowering the power setpoint of the gas turbine and the decrease in the actual power. Secondly, grid code requirements and the cycle time of the control system result in a maximum reaction time of up to 250 ms before adjusting the positions of the fuel gas control valves. Both delays have an influence on the maximum rotational speed (see Fig. 2), but the reaction time of the control system is a fixed parameter. Once the control system has adjusted the corresponding parameters, the power of the gas turbine decreases and the rotor speed as the effective power output of the gas turbine turns negative (see Figs. 1 and 2).

The increase in the rotational speed must – of course – not exceed the mechanical limits of the shaft. Therefore, load shedding events are simulated with different scenarios (e.g., normal load shedding, clamping of the control valve and freezing of the controller) with every update of a gas turbine to identify worst-case scenarios, i.e., finding the maximum overspeed value for all possible combinations of the corresponding parameters within the respective ranges, and ensure that the mechanical integrity is not violated.

The resulting maximum rotational speed for a load shedding event is influenced by several parameters (see Eq. 3), e.g., the state of the fuel supply system, ambient conditions and – of course – the current mechanical power of the turbine in the moment of the event. For a complete overspeed analysis, around 10 parameters can be considered. As mentioned previously, the simulation deals with a complex model. The overspeed as an objective function is therefore described as a black-box function with the following characteristics:

$$n_{max} = f_{blackbox}(p_{gas}, H_{ux}, t_{gas}, P_{setpoint}, n_{trip}, n_0, t_a \quad (3)$$

where p_{gas} is the fuel gas supply pressure, H_{ux} is the lower heat value of the fuel gas, t_{gas} is the fuel gas temperature, $P_{setpoint}$ is the load setpoint for the turbine, n_{trip} is the trip limit for rotational speed, n_0 is the net frequency at the time of load shedding *and* t_{amb} is the ambient air temperature.

Models for the simulation of the turbine operation include the thermodynamics and the control system and are therefore complex. The simulation for a single set of parameters takes up to 20 min. Computing a reasonably high number of variations of influencing parameters for an overspeed analysis may require several thousand simulations and therefore requires an enormous runtime.

Applying methods of BBO to this problem, the required number of simulations can be reduced because the methods will search for the actual optimum, instead of covering the whole range of variables in a constant resolution. Therefore, the analysis can be executed faster. For this work, a simplified overspeed analysis was performed by BBO methods. The resulting maximum speed of the gas was computed only for varying conditions of the fuel itself, namely, lower heat value H_{ux}, fuel gas temperature T_{gas} and fuel gas supply pressure p_{gas}. Furthermore, only normal load shedding was simulated without additional failure scenarios.

2.2. Use Case 2: transfer function identification

As a second use case, transfer function coefficients for a delayed ramp function (ramp function applied to a lag element) are to be identified. The behaviour of a delayed ramp is used to approximate the mass flow supplied to a gas turbine during a load shedding event when the fuel gas valves are closed instantly.

The physical background is simple: to reduce the fuel gas mass flow, the corresponding valve is moved from a certain position for the steady-state operation to its closed position. This movement can me modelled as a ramp. Between the valve and the CC, where the fuel gas is finally converted into thermal energy, the fuel gas must flow through a piping volume (see Fig. 3). This volume dampens the reduction of the mass flow rate. Furthermore, this volume is under a higher pressure than the CC. Due to the pressure compensation, fuel gas will flow into the CC from the piping for a short time after the valve is completely closed. This results in the dynamics of the mass flow rate, as shown in Fig. 4.

DOI 10.2478/ama-2023-0034
acta mechanica et automatica, vol.17 no.2 (2023)
Special Issue "Modeling and experimental investigations of thermo-hydraulic, hydrogen, manufacturing and mechanical systems"

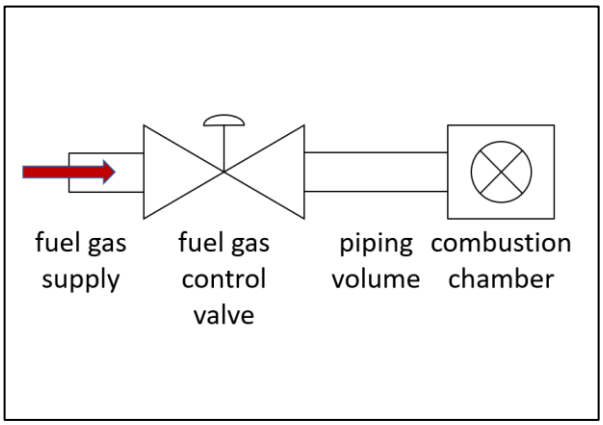


Fig. 3. Schematic fuel gas piping

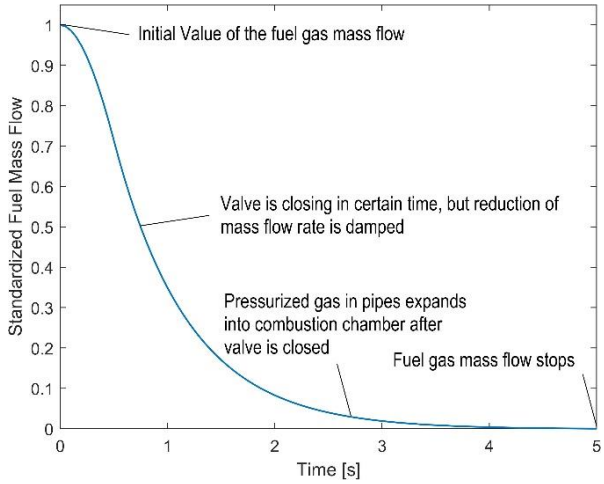


Fig. 4. Decrease in the supplied fuel mass flow during and after valve closing

As a reference value used to compare the results of the BBO methods, a minimum error can be manually determined as follows:

A lag element can be described with the following differential equation:

$$T \cdot \dot{y}(t) + y(t) = K \cdot u(t). \tag{4}$$

where u is the input of the lag element and y is the output of the lag element.

Integration and rearranging of this equation give the following (with K assumed to be included in $u(t)$):

$$T[(y(t) - y_0)] = U(t) - Y(t) \tag{5}$$

where U and Y are antiderivatives of u and y for $U(0) = Y(0) = 0$.

The input of the lag element is separated into different functions: a ramp function from $t = 0$ to $t = T_r$. At T_r, the ramp has reached its final level and is from here on constant (in the case discussed here, it is 0). A ramp is defined as follows:

$$u_1(t) = (y_1 - y_0) \cdot \frac{t}{T_r} + y_0 \quad \text{for } 0 \le t \le T_r \tag{6}$$

where y_0 is the start value of a ramp, y_1 is the end value and T_r is the time between the start and the end.

As mentioned, the value of the input function after T_r is constant with a value of y_1:

$$u_2(t) = y_1 \quad \text{for } t \ge T_r \tag{7}$$

Given a set of n measured points of the real behaviour $[t_i, y(t_i)]$ for $i = 1$ to n, a system of linear equations can be set up and solved using the least-squares method. Given a set $[T, T_r]$, an approximated value y_{ap} for the corresponding lag element can be calculated for each t_i and the error between the measured and the calculated value can be taken as the sum of squares of the errors for each point t_n

$$\Delta y(T_r, T) = \sum_{i=1}^{n} (y_{ap}(T_r, T, t_i) - y(t_i))^2 \tag{8}$$

for any given set of pairs $[t_i, y(t_i)]$.

For this application, it is also possible to calculate the time coefficients T_r and T in a more direct manner, with the following vectors:

$$\begin{aligned} \vec{y} &= (y_1, y_2, \dots, y_n)^T \\ \vec{U} &= (U_1, U_2, \dots, U_n)^T \\ \vec{Y} &= (Y_1, Y_2, \dots, Y_n)^T \end{aligned} \tag{9}$$

Eq. 5 provides a value for T with a given value T_r. With this, it is possible to minimise Eq. 8 only as a function of T_r and determine T afterwards.

Identifying functions like this is necessary to develop simplified models of gas turbine power plants that are, e.g., used in studies to analyse the interaction with the electrical grid.

Basically, this is not a black-box function as these values can be calculated numerically, and furthermore, in terms of computational expenses, it is very cheap. Nevertheless, this problem can serve as a simple version of the identification of more complex models and shows how the performance differs between certain levels of complexity.

3. BLACK-BOX OPTIMISATION

When explaining BBO, it is necessary to define two terms: black-boxes and optimisation.

Black-box. A black-box is, according to Kimiaei and Neumaier, "[…] an oracle that returns for a given x ∈ $\mathbb{R}^n$ the function value f(x)" [1] (p. 1). The main characteristic is the non-visibility of the inner computational process of the function. Applying a set of input variables to a black-box function, it will return a (set of) output variable(s), and there is no way to investigate the procedure of computing this output with reasonable effort.

Optimisation: From a mathematical point of view, optimisation is simply the determination of – local or global – minima and maxima of a given function. Commonly, this calculation is driven by derivatives of the function.

Combining these two definitions, we need to find those optima without having a real insight into the function and therefore no information about its derivatives.

Over the past decades, various methods for BBO have been introduced for a variety of applications, but a universal algorithm that can be applied to any problem is not developed yet [7] (pp. 123–124).

There are several categorisations for BBO or DFO methods, and in this study, the categorisation suggested by Rios and Sahinidis [3] was used. They divide algorithms based on the following properties:

- Direct methods or model-based methods
  The (expensive) objective function can either be evaluated itself or an approximation (or surrogate) model can be used for optimisation.
- Global or local methods
  Local methods are designed to quickly find the next local optimum but miss the capability to check if there exist other optima, whereas global methods will have this capability and will determine an absolute optimum.
- Stochastic or deterministic methods
  Categorisation by a deterministic or randomised determination of the algorithm's search steps.

In addition to Rios and Sahinidis [3], Larson, Menickelly and Wild [7] have also published a profound review of methods for DFO and BBO, where examples for each category are given.
Furthermore, methods differ in their ability to deal with boundaries of the search space. They might require boundaries in the form of equality or inequality constraint, while others cannot process any limitation of the input variables.

Of course, there also exist numerous methods that are hybrids and combine the categories mentioned before.

4. SELECTED BBO ALGORITHMS

For the use cases to be addressed in this study, there are several conditions of the objective functions that the optimisers will have to cope with. These are listed as follows:

- Noise
  The models used might contain stochastically influenced functions, and therefore, the same input can provide a slightly different simulation output.
- Boundaries
  Input variables will mostly have lower and upper limits. The highest or lowest values might therefore lie on these boundaries, instead of forming an optimum within the search region.
- Local Optima
  Due to the interdepending influence of different parameters on the objective function, multiple local optima might occur. For most cases, a global optimum or the highest/lowest function value within a search region is important to be found, and therefore, the optimisation should not stop at a local optimum.

The existence of noise is applicable for both use cases mentioned: in the operation simulation for the overspeed analysis, the simulation of measurements within the gas turbine model is modified with artificial noise as they would be noisy in the operation of a real plant. For use case 2, we have to assume that the measured flow rates used to identify the transfer function are provided with usual measurement errors. The fuel gas parameters for the overspeed analysis are limited to the fuel definition of a contract for a plant. The boundaries for the parameters of the transfer function are, in this case, chosen by the previous numerical calculation of the problem. For a totally unknown transfer function, the upper limits would have to be estimated by the user, while the lower limits could always be set to 0. Concerning the existence of multiple local optima, the overspeed analysis could have multiple local tallest points due to the interdependency of the variables. For use case 2, the previous evaluation has shown that only one optimum exists, but considering that the methods should be applied to black-box functions with an unknown structure, the existence of local optima should always be considered.

Taking these conditions into account and based on the nomenclature provided by Rios and Sahinidis [3], only global optimisation algorithms are able to appropriately solve these problems. Additionally, all methods that do not accept boundaries cannot be considered.

Furthermore, in terms of usability, only few algorithmic parameters of a BBO method should be needed to be adjusted for any new problem.

To evaluate how well BBO methods perform applied to the exemplary problems (Section 2), methods that are available as open source and, at best, as MATLAB® software will be tested preferably. Two algorithms that meet the conditions outlined previously are the Multilevel Coordinate Search (MCS) algorithm [20] and the Stable Noisy Optimization by Branch and Fit (SNOBFIT) algorithm [21] by Huyer and Neumaier. A different algorithm that is a classically model-based approach is the blackbox algorithm [22] by Knysh and Korkolis. This algorithm is available as a Python® package and translated to MATLAB® for this study by the authors.

The MCS algorithm is a direct evaluation method, whereas SNOBFIT uses different local models, and blackbox uses a global model of the objective function; therefore, they cover a broad range of the algorithmic spectrum. Additionally, MCS and SNOBFIT showed satisfactory results in the review of Rios and Sahinidis [3] (pp. 1267–1287).

The concepts of the algorithms are briefly presented in the following paragraphs; for a complete description of all details, we refer to the corresponding papers noted previously.

4.1. Global optimiSation by MCS

The so-called MCS algorithm was developed by Waltraud Huyer and Arnold Neumaier in 1998 [20]. The search space is continuously divided into boxes that have a base point and a corresponding function value given by the objective function. Partitioning of the boxes is executed along the coordinate axis, and with each splitting of a box, a level s is increased. Any box can be split until this level reaches the level $s = s_{max}$, which indicates that a box is too small for another division [3] (p. 1253).

Fig. 5 shows a two-dimensional search space that has already been split several times (black lines). The point in the middle (green-yellow) provides the best function value. The rectangle highlighted in magenta is chosen to be split next, and the green-highlighted point (approximately at [0.5; 1.5]) marks the position for the next split.

The algorithm selects the boxes to be divided in the next iteration by the values of the objective functions of the base points (see Figs. 5 and 6). There are two options for splitting the boxes:

Splitting by rank: Boxes already having a high level of s can be chosen for splitting by rank. This means that the coordinate along which they will be divided will be chosen by how many splits have already been performed in each coordinate. Selecting the coordinate with the lowest numbers helps decrease unexplored spaces in the corresponding direction.

Splitting by expected gain: The second method for splitting is used for lower levels of *s*. Then, a local quadratic model of the objective function is built using points from previous iterations, and the splitting coordinate is chosen by optimisation of this model.

Additionally, local searches can be executed when a box has already reached the splitting level s_{max}.

As seen in Fig. 6, the algorithm has reached a higher density within a region of low function values (dark green), which means that in this case, the boxes were split by the expected gain. While proceeding the search, promising areas will be explored more and more. The convergence to a global minimum is ensured when s_{max} approaches to infinity, which means that the number of functions evaluations is not limited [20] (pp. 347–349).

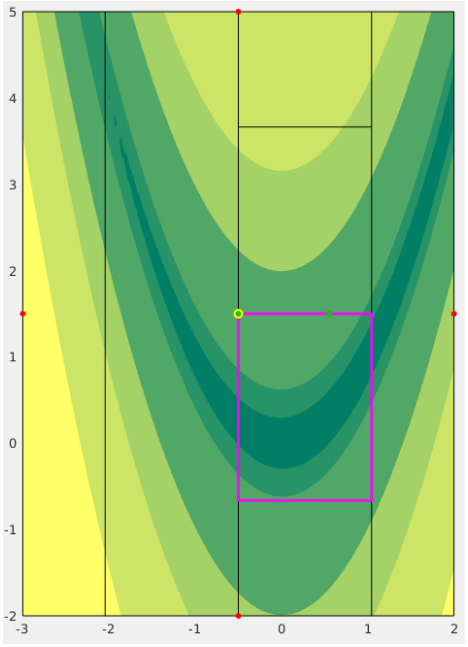


Fig. 5. Rectangle chosen to be split by the MCS algorithm marked in magenta [Figure: "MCS algorithm" by Najko32. License: CC BY-SA 4.0] [26]. MCS, multilevel coordinate search

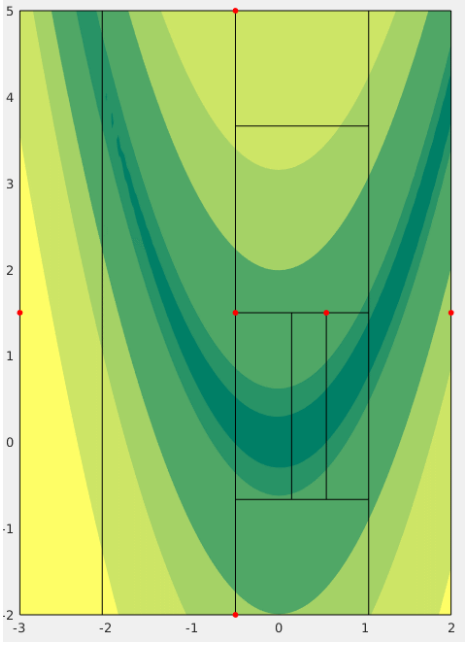


Fig. 6. Result of the iteration of the MCS algorithm based on Fig. 5 [Figure: "MCS algorithm" by Najko32. License: CC BY-SA 4.0] [26]. MCS, multilevel coordinate search

4.2. Stable noisy optimisation by branch and fit

Huyer and Neumaier [21] have published another promising algorithm for BBO: SNOBFIT. The sequence of the SNOBFIT algorithm is as follows:

Initialisation: Given a set of points within the search space and the objective function, the initial setup for the algorithm is built.

Consecutive iterations: Based on this set, SNOBFIT generates the user-defined number of required new evaluation points n_{req}. The search space is split into boxes so that each contains exactly one point. A smallness is computed for each box to have a measure on how often a box has already been split, and a local fit around each point is computed (or updated for old points). The split is performed along the coordinate axis that shows the highest variance of the points in the box to be split. The current best point and the corresponding function value are evaluated, and a local quadratic fit around this point will be computed. Points to be evaluated are prioritised in different classes:

- Class 1: By optimisation of the aforementioned fit, a new potentially best point can be generated.
- Class 2: Points determined by a trust region estimation around the so-called local point that shows a significantly lower function value than its nearest neighbours.
- Class 3: Points determined by a trust region estimation about non-local points.
- If less than n_{req} points have been chosen, two more classes of points can generated.
- Class 4: Points that split large boxes to explore uncovered areas of the search space.
- Class 5: Points determined by a randomised space-filling set of the missing number of points are created in a way that these points have the maximum distance to all other points.

All generated points are evaluated by the objective function, and a new iteration will start.

Stopping. The algorithm stops if for a defined number of consecutive iterations, no new point of class 1 has been generated, and the current best point is therefore assumed to be the global optimum.

4.3. Blackbox: a procedure for parallel optimisation of expensive black-box functions

Knysh and Korkolis [22] have provided a method for BBO that utilises a global response surface with radial basis functions (RBFs). The method is originally provided as a Python® package [23] and was implemented in MATLAB® for this study.

For the procedure, all variables are rescaled to a range of [0, 1] so that the search space equals a unit hypercube. Furthermore, after initial sampling, the function values are also rescaled to a range from 0 (best or lowest function value) to 1 (worst or highest function value).

The initial sampling set is generated by quasirandom sequences [24]. After the first approximation of the response surface is calculated, further sampling candidates are provided by an adapted version of the Constrained Optimization using Response Surfaces (CORS) algorithm [25]:

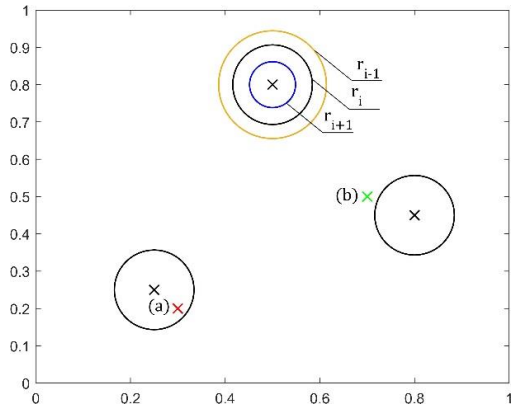


Fig. 7. Demonstration of the blackbox algorithm

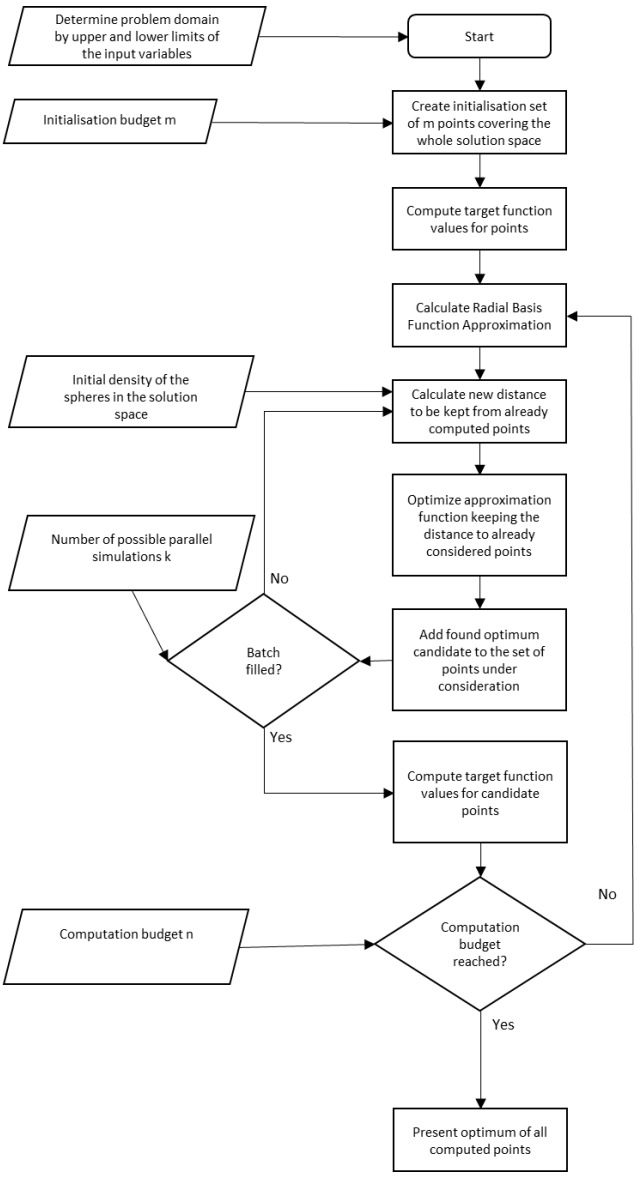


Fig. 8. Flowchart of the blackbox algorithm

Basically, the RBF approximation is updated in each iteration step by adding candidate points for the global optimum that minimise the fitted function under the following constraint: the new candidate must lie outside of a (hyper-)sphere around all other points (already sampled and current candidates, marked with a black x in Fig. 7), which is also shown in Fig. 7: the point (a) highlighted in red lies within the current radius r_i and is therefore not used as a new candidate, whereas the green point (b) meets the distance constraint. The radius of the spheres (i.e., the distance a new point must keep to all other sample points) decreases over the number of iterations (see r_{i-1}, r_i and r_{i+1} in Fig. 7). By this, the algorithm first explores the whole search space and then proceeds to a more local search. The speed of decreasing (see Fig. 8) is set by the user.

The algorithm stops after a user-specified number of function evaluation is reached (Fig. 8). Even though the authors do not mention any proven convergence of the algorithm, it will cover the search space densely, if the number of function evaluations is not limited and the values for the initial density (defining the starting value for the spheres' radii) and its decrease are chosen to focus on a global search. Thereby, the algorithm will determine a global optimum.

5. APPLICATION OF THE METHODS TO CHOSEN USE CASES

All algorithms are applied to both use cases. As the first of both applications (Section 2) – the gas turbine overspeed analysis – is expensive in terms of computation, only one run per algorithm is performed, whereas the identification of the transfer function is solved 20 times to see whether the quality of the solution found after varies between the executions of the code or not. To compare the algorithms, the following measures are considered:

- Number of function evaluations used to determine the global maximum
- Execution time for a single run of the algorithm
- Quality of the found global optimum function value
- Quality of the found optimum input variables

Each method is limited to about 200 function evaluations. As a benchmark for the real maximum of the overspeed value, a grid search is performed, that is – compared to how the grid search is performed in practice – considerably denser. In this manner, we aim to obtain a point as close as possible to the actual maximum for the overspeed value. To validate how the BBO algorithm performs in comparison to the standard procedure, another grid search[1] based on 216 equally distributed evaluation points was executed. This run will be referred to as "sparse grid search," whereas the formerly mentioned will be called "dense grid search."

If parallelisation is possible, the number of parallel function evaluations is set to 8. By this limit, the RAM of the testing computer works almost at capacity during the compilation process but does not exceed it, which would result in out-of-memory errors.

Additionally, the resolution vector for SNOBFIT was set to [H_{ux}: 1 MJ/kg, T_{gas}: 1°C, p_{gas}: 1 bar] (see Section 2.1). The resolution is based on the practical use: a) for each variation of the heat value, a new compilation of the model is necessary. For this reason, the number of possible values needs to be limited and

[1] In a grid search, a set of certain values to be considered for simulation is defined for each input variable. Combining the sets of all variables provides a grid covering the whole search space in a certain resolution. All points of the grid will be used to compute a value of the objective function and the optimum of these values is returned as the determined global optimum.

sciendo
DOI 10.2478/ama-2023-0034
acta mechanica et automatica, vol.17 no.2 (2023)
Special Issue "Modeling and experimental investigations of thermo-hydraulic, hydrogen, manufacturing and mechanical systems"

b) the overspeed value does not change substantially within the space between the values allowed by the resolution vector, and furthermore, the values will not be stated in higher accuracy in the documentation for customers.

The results and corresponding parameters are given as percentages of the values for the benchmark results to simplify the comparison.

The objective function is covered in a small MATLAB® function that manages the parallelisation of simulations, if necessary, and returns the overspeed values for the points requested for an iteration step by the algorithms.

For use case 2 (parameter identification of a transfer function, Section 2.2), the averaged values over all 20 runs are taken for evaluation of the performance. The variation of the results is compared separately. SNOBFIT and blackbox were able to compute four function values at the same time. Furthermore, the resolution vector for SNOBFIT was set to [t_r: 10^{-6} s; T: 10^{-6} s] as the standard search has shown that further quantisation does not substantially improve the found minimum but – of course – increases the number of calculations needed. Other parameters do not differ between use cases 1 and 2. Again, to run the algorithms, a simple function that returns the error values corresponding to the requested points and managing the parallelisation is sufficient.

5.1. Results for gas turbine overspeed analysis

For the gas turbine overspeed analysis, the objective functions are defined as a black-box function, as shown in Eq. 3. For the evaluation of the capability of BBO to solve problems based on computationally expensive simulations, the objective was reduced on the following parameters:

$$n_{max} = f_{blackbox}(p_{gas}, H_{ux}, t_{gas}) \tag{10}$$

Obviously, the most important question to answer is whether the methods are capable of providing a value sufficiently close to the reference value (dense grid search) for the objective function value (namely, the maximum overspeed value). Fig. 9 displays how close the computed optimum values are to the reference value determined by a dense grid search with multiple thousand simulations covering the search space. All methods provide a value that differs less than 0.15% from the reference optimum and therefore meet the previously mentioned requirement. Among these overall satisfactory results, blackbox provides an optimum value that is – considering noise – equal to the reference value.

Nevertheless, it is also relevant if the suggested optimum is close to the actual optimum in terms of coordinate values. Fig. 10 shows the deviation of the found optimum from the reference values for all methods. Noticeably, the divergence in the fuel gas supply temperature T_{gas} is enormous. By contrast, SNOBFIT, blackbox and the sparse grid search deliver values for the heat value H_{ux} and the fuel gas supply pressure p_{gas} that are close to the actual optimum coordinates. It is thereby clear that T_{gas} has only a minor influence on the maximum rotor speed.

Additionally, the points suggested by SNOBFIT and blackbox are both only insignificantly different from points within the top 5 points (best 0.15%) of the dense grid search and therefore extremely close to the reference optimum point. As MCS provides the worst (even if still good) value for the maximum rotor speed, the determined values for the input variables are also the farthermost from the reference optimum from the dense grid search.

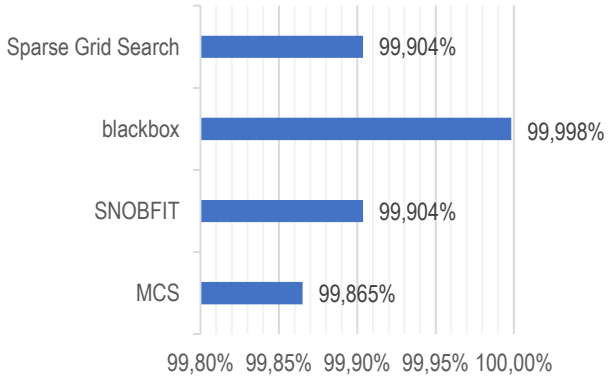
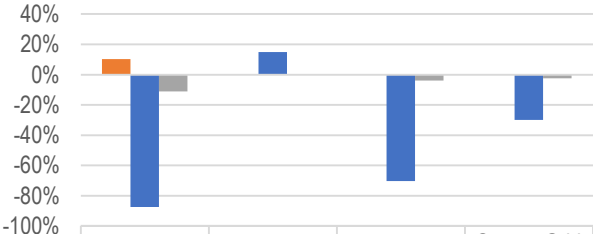


Fig. 9. Accuracy of determined optimum for BBO methods for use case 1. BBO, black-box optimisation; MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit



	MCS	SNOBFIT	blackbox	Sparse Grid Search
Hux	10,34%	0,00%	0,00%	0,00%
Tgas	-87,50%	15,00%	-70,24%	-30,00%
Pgas	-11,09%	0,00%	-3,92%	-2,50%

Fig. 10. Accuracy of optimum coordinates of BBO methods for use case 1. BBO, black-box optimisation; MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit

Provided that all methods compute adequate values for the value of interest, it is of same importance how efficient they find the corresponding solution. In Fig. 11, the numbers of function evaluations needed to determine the optimum are compared for the three tested BBO methods and the Sparse Grid Search as a reference to the currently used procedure. As already mentioned, the BBO methods were limited to about 200 iterations. Therefore, the grid search was designed to have a comparable computational effort.

While blackbox and MCS use all possible iterations (or due to local searches even more), SNOBFIT stops the iterations after 64 function evaluations after it has not been able to improve the current optimum for five iterations.

In this application, the computational expense between two simulations might differ significantly. The used Simulink® model is compiled to C-code, and an .exe file is built to run the actual simulation. Depending on which parameters change, the .exe file can be used again, and the model does not need to be compiled again. The actual simulation runs only few minutes on the system used, whereas the compilation might take up to 15 min. In use case 1, a change of the heat value requires a new compilation.

Fig. 12 shows the runtime of each method and, in this case, also for the dense grid search. Most remarkable is the runtime of MCS: it finishes the run with 225 function evaluations after more

DOI 10.2478/ama-2023-0034

than 1 day and 8 h. By contrast, blackbox does only need about 7 h, whereas SNOBFIT needs a runtime of 1 h and 44 min, which is very close to the reference method, which needs 1 h and 14 min. The explanation for this is simple:

Firstly, MCS changes the coordinates of requested points only slightly, which results in a very high number of recompilations. Contrarily, both grid searches and SNOBFIT have a limited resolution of the axes and therefore provoke only few compilations. As the sparse grid search does only compile seven times, whereas SNOBFIT and the dense grid search need 21 compilations, it is faster than SNOBFIT even though it takes many more iterations.

The second obstacle for MCS is its seriality: it cannot be parallelised, while all other algorithm work with eight simulations at each iteration. This is the main reason that blackbox performs better than MCS even if it needs the same number of compilations.

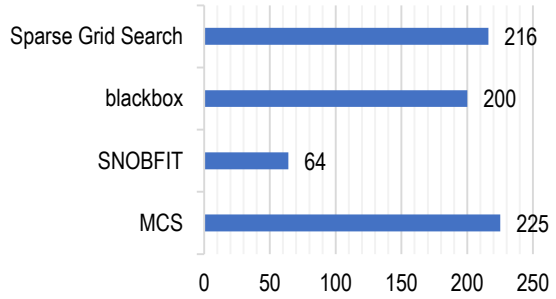


Fig. 11. Number of functions evaluations needed to solve use case 1. MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit

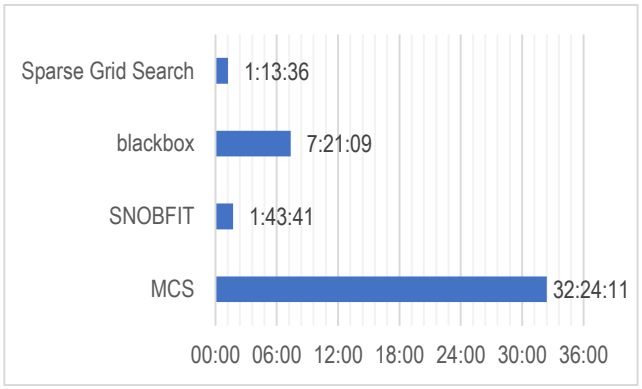


Fig. 12. Runtime of BBO methods needed to solve use case 1. BBO, black-box optimisation; MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit

This use case with a complex or computational expensive objective function brings up three important learnings:

- All methods provide sufficient solutions for the given problem.
- The performance of an algorithm strongly depends on the degree of parallelisation.
- A reasonably coarse variation of input parameters is extremely beneficial.

The following second use case will show if these aspects are also relevant for less complex and expensive objective functions.

5.2. Results for transfer function identification

Again, the first point of interest is how precise the identified optima are compared to the reference value. As mentioned earlier, for this application, the actual solution can be computed numerically. Tab. 1 shows the results of the three algorithms:

Tab. 1. Results for use case 2

Name	Standard	MCS	SNOBFIT	blackbox
Minimum error determined	0.229974	0.229977	0.230002	0.230126
Deviation from standard value	-	0.0011%	0.0123%	0.0658%
***Tr* for minimum error**	0.11893	0.11897	0.11895	0.11884
Deviation from standard value	-	0.0326%	0.0167%	-0.0731%
***T* for minimum error**	0.426205	0.426207	0.426212	0.426029
Deviation from standard value	-	0.0005%	0.0015%	-0.0414%

MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit.

It is clearly visible that all algorithms determine the optimum extremely close to the actual values in terms of the least error as well as in terms of the variables tr and T. Nevertheless, it is remarkable that for this use case, MCS delivers the most accurate optimum value and also very precise parameters. Even though both provided solutions are still more than acceptable, MCS is about 10 times more precise than SNOBFIT and about 60 times more precise than blackbox.

Having such close results, it is even more important to relate these results to the computational performance, which is shown in Figs. 13 and 14.

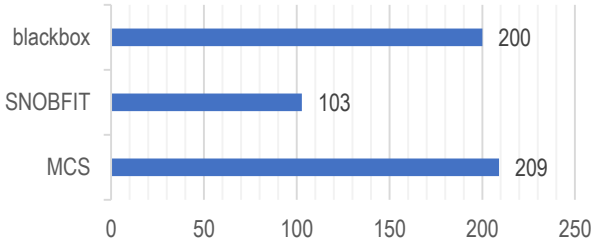


Fig. 13. Number of function evaluations needed to solve use case 2. MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit

Fig. 12 shows that while blackbox and MCS consume the whole computational budget, SNOBFIT finds the sufficient solution only using half the function evaluations.

Fig. 14 shows the average time the methods needed to execute use case 2 in the test. A clear ranking can be derived from these numbers: SNOBFIT also outperforms the other

algorithms by solving the problem in only 5 s, while MCS needed about 12 s, and blackbox is even more slow, with 30 s.

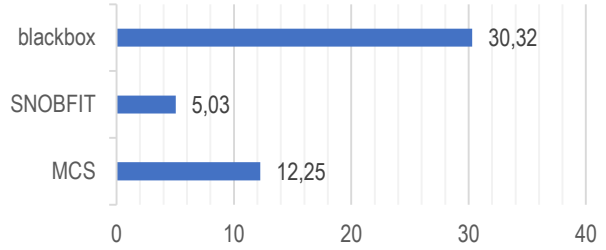


Fig. 14. Runtime of BBO methods [s] needed to solve use case 2. BBO, black-box optimisation; MCS, multilevel coordinate search; SNOBFIT, Stable Noisy Optimization by Branch and Fit

In terms of replicability of the results, MCS and blackbox only differ in the runtime over the 20 runs (MCS: minimum runtime: 11.61 s / maximum runtime: 13.78 s; blackbox: 29.15 s / 37 96 s), and SNOBFIT's performance is more volatile: the least number of function evaluations to find the optimum for the transfer functions parameters was 71, and the highest number of evaluations was 147 and the runtimes vary accordingly between 3.35 s and 6.81 s. Nevertheless, the quality of the found minimum error is only varying in the range about ±0.1%, and the actual parameters do not fluctuate more than 0.4% around the average values.

Even though SNOBFIT varies in its efficiency, the worst performance is still significantly above the best performances of blackbox or MCS.

For this use case, all methods are fully competitive to the reference value. SNOBFIT has clear advantages in terms of computational and time efficiency, whereas MCS is more precise. The significantly higher runtime of blackbox might not be dramatic in absolute terms but is still a remarkable disadvantage.

6. CONCLUSIONS

Three BBO methods (MCS, SNOBFIT and blackbox) were tested on two use cases with very high and very low computational expense. The application of the methods to our exemplary problems have been proven to be highly user-friendly: setting up a sufficient MATLAB® function to be called and the maximum number of function calls, and all methods run stable and provide satisfactory results. Only for SNOBFIT, the input of the resolution vector of the input variables is an additional value to be specified.

All methods provided optima for the objective function close to the set reference values and do therefore meet the most basic requirement.

Additionally, all coordinate values determined to be the optimum point were close to the actual points from the reference optimum. Even though MCS cannot compete in accuracy for the overspeed analysis, it is the most precise algorithm for the identification of the transfer function parameters.

In summary, the accuracy of the found optimum is satisfying for all algorithms in both applications out of common power engineering tasks, and therefore, methods of BBO can be considered for solving these kinds of problems as questioned in Chapter 1.

To be competitive with the procedures used currently, the computational efficiency (actual runtime of the algorithms) is nearly as important as the quality of the found solution. The runtimes for the identification of transfer function parameters (use case 2, Section 2.2) differ significantly but are still short in absolute terms. Still, it is to mention that SNOBFIT is significantly faster than MCS, and blackbox is – compared to this – slow in this use case. It is – of course – still much more efficient than the manually performed procedure to find the best fitting parameters.

Of course, the time efficiency is far more important when dealing with more expensive objective functions as the overspeed analysis. Here, MCS and blackbox cannot compete with the sparse grid search and SNOBFIT. The foundation for that discrepancy is the compilation of the models to be simulated, as described in Section 5.1.

Furthermore, the results point out that parallelisation is necessary for complex problems.

Referring to the questions addressed at the beginning of this article, it was shown that BBO methods are easily applicable to simulation studies and that they do have the potential to meet the tasks arising in the context of gas turbine control simulation. However, for computationally expensive functions, only SNOBFIT is, in general, competitive to a knowledge-based grid search as currently performed as a standard procedure, and it also shows the best performance for a cheap objective function.

7. OUTLOOK

While the methods tested for this study already show promising results, there are still several questions that should be answered in future work.

Firstly, the whole range of BBO algorithm has not been tested yet, and especially the field of genetic or evolutionary algorithms, which has seen a significant development in the past years is not included and should be considered in further studies. Currently, there are ongoing studies on the implementation of additional algorithm, which will be tested soon. The experience portrayed in this work is promising that the adaption of further methods will be successful.

Secondly, it was clearly shown that for the gas turbine overspeed analysis, the high number of necessary compilations is limiting the performance of the BBO methods, blackbox and MCS. The performance of SNOBFIT for this application is far more effective, which is based on less compilations due to the resolution vector. Therefore, we aim to adapt this resolution vector to all methods.

For a non-simplified overspeed analysis with more input variables, the BBO methods might be significantly more competitive to the standard grid-based procedure as it will perform far less efficient as the number of points increases exponentially with the dimensions. This results either in a worse accuracy or in a far longer runtime.

Considering that most parameters to adjust the algorithms' behaviour have been kept to standard values, as provided in the corresponding literature, the performance might be improvable. Combining this with the idea of the resolution vector for the input variables, the performance of all methods will possibly increase. Still, it needs to be ensured that an adjustment of the algorithmic parameters works properly for all use cases and does not provide benefits just for one problem.

Furthermore, the actual usability and universality must be tested on more applications. Studies testing the methods for the identification of a complex gas turbine model with real-world data and the adjustment of control parameters will follow.

REFERENCES

1. Kimiaei M, Neumaier A. Efficient Global Unconstrained Black Box Optimization. Mathematical Programming Optimization. 2022;*14*: 365-414. https://doi.org/10.1007/s12532-021-00215-9
2. Custódio AL, Scheinberg K, Vicente LN. Methodologies and Software for Derivative-free Optimization. In Advances and Trends in Optimization with Engineering Applications (SIAM). 2017: 495-506. https:///doi.org/10.1137/1.9781611974683.ch37
3. Rios LM, Sahinidis NV. Derivative-free optimization: a review of algorithms and comparison of software implementations. Journal of Global Optimization. 2013;*56:* 1247-1293. https://doi.org/10.1007/s10898-012-9951-y
4. Larson J, Menickelly M, Wild SM. Derivative-free Optimization Methods. Acta Numerica. 2019;*28:* 287-404. https://doi.org/10.1017/S0962492919000060
5. Amaran S et al. Simulation Optimization: A Review of Algorithms and Applications. Ann Oper Res. 2016;240: 351–380. https://doi.org/10.1007/s10479-015-2019-x
6. Ammeri A, Hachicha W, Chabchoub H, Masmoudi F. A comprehensive literature review of mono-objective simulation optimization methods. Advances in Production Engineering & Management. 2011;6(4): 291–302.
7. Walton S, Hassan O, Morgan K. Selected Engineering Applications of Gradient Free Optimisation Using Cuckoo Search and Proper Orthogonal Decomposition. Archives of Computational Methods in Engineering. 2013;20: 123-154. https://doi.org/10.1007/s11831-013-9083-7
8. Yang XS, Deb S. Engineering Optimisation by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation. 2010;1(4): 330–343. https://doi.org/10.48550/arXiv.1005.2908
9. Xing XQ, Damodaran M. Assessment of Simultaneous Perturbation Stochastic Approximation Method for Wing Design Optimization. Journal of Aircraft. 2002;39: 379–381. https://doi.org/10.2514/2.2939
10. Xing XQ, Damodaran M. Application of Simultaneous Perturbation Stochastic Approximation Method for Aerodynamic Shape Design Optimization. AIAA Journal. 2005;43(2): 284–294. https://doi.org/10.2514/1.9484
11. Xing XQ, Damodaran M. Inverse Design of Transonic Airfoils Using Parallel Simultaneous Perturbation Stochastic Approximation. Journal of Aircraft. 2005;42(2): 568–570. http://dx.doi.org/10.2514/1.10876
12. Kothandaraman G, Rotea MA. Simultaneous-Perturbation-Stochastic-Approximation Algorithm for Parachute Parameter Estimation. Journal of Aircraft. 2005;42(5): 1229–1235. http://dx.doi.org/10.2514/1.11721
13. Prakash P et al. Design Optimization of a Robust Sleeve Antenna for Hepatic Microwave Ablation. Physics in Medicine and Biology. 2008;53: 1057–1069. https://doi.org/10.1088/0031-9155/53/4/016
14. Li Y. A Simulation-based Evolutionary Approach to LNA Circuit Design Optimization. Applied Mathematics and Computation. 2009; 209(1): 57–67. http://dx.doi.org/10.1016/j.amc.2008.06.015
15. Radac MB et al. Application of IFT and SPSA to Servo System Control. IEEE Transactions on Neural Networks. 2011;22(12): 2363–2375. https://doi.org/10.1109/tnn.2011.2173804
16. Ernst D et al. The Cross-Entropy Method for Power System Combinatorial Optimization Problems. Power Tech. IEEE. 2007: 1290–1295. https://doi.org/10.1109/PCT.2007.4538502
17. Kowalczyk Ł, Elsner W, Niegodajew P. The Application of Non-Gradient Optimization Methods to New Concept of Power Plant. 6th IC-EpsMsO; 2015 Jul 8-11; Athens.
18. Lu S. Dynamic modelling and simulation of power plant systems. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy. 1999;*213(1)*: 7-22. https://doi.org/10.1243/0957650991537392
19. Huan J et al. The Application of Digital Twin on Power Industry. IOP Conf. Series: Earth and Environmental Science. 2021;647. https://doi.org/10.1088/1755-1315/647/1/012015
20. Huyer W, Neumaier A. Global Optimization by Multilevel Coordinate Search. Journal of Global Optimization. 1999;14(2): 331-355. https://doi.org/10.1023/A:1008382309369
21. Huyer W, Neumaier A. SNOBFIT - Stable noisy optimization by branch and fit. ACM Transactions on Mathematical Software. 2008; *35(2):* Article No.: 9, 1-25. https://doi.org/10.1145/1377612.1377613
22. Knysh P, Korkolis Y. blackbox: A procedure for parallel optimization of expensive black-box functions. *arXiv (cs.MS)*. preprint submitted 2016, https://doi.org/10.48550/arXiv.1605.00998
23. Knysh P. blackbox: A Python module for parallel optimization of expensive black-box functions [Internet]. [place unknown]; [publisher unknown]; 2016 Feb 19 [updated 2022 Sep 5; cited 2021 Oct 17]. Available from: https://github.com/paulknysh/blackbox
24. Roberts M. Extreme Learning. The Unreasonable Effectiveness of Quasirandom Sequences [Internet]. [place unknown]; [publisher unknown]; 2018 Apr 25 [cited 2022 June 2]. Available from: http://extremelearning.com.au/unreasonable-effectiveness-of-quasirandom-sequences/
25. Regis RG, Shoemaker CA. Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions. Journal of Global Optimization. 2005;*31*: 153-171. https://doi.org/10.1007/s10898-004-0570-0
26. Najko32. MCS algorithm [Internet]. [place unknown]; [publischer unknown]; 2018 August 6 [cited 2023 February 9]. Available from: https://commons.wikimedia.org/wiki/File:MCS_algorithm.gif
27. Nelder JA, Mead R. A Simplex Method for Function Minimization. The Computer Journal. 1965;7: 308-313. https://doi.org/10.1093/comjnl/7.4.308
28. Winfield D. Function Minimization by Interpolating in a Data Table. IMA Journal of Applied Mathematics. 1973;12: 339-347. https://doi.org/10.1093/imamat/12.3.339
29. Holland JH. Genetic Algorithms and Adaptation. Adaptive Control for III-Defined Systems. 1984: 317-333. https://doi.org/10.1007/978-1-4684-8941-5_21
30. Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and Analysis of Computer Experiments. Statistical Science. 1989;4: 409-423. https://doi.org/10.1214/ss/1177012413
31. Jung I et al. Computational Fluid Dynamics Based Optimal Design of Guiding Channel Geometry in U-Type Coolant Layer Manifold of Large-Scale Microchannel Fischer–Tropsch Reactor. *Ind. Eng. Chem. Res.* 2016;55: 505–515. https://doi.org/10.1021/acs.iecr.5b03313
32. Hemmat Esfe M, Hajmohammad M, Moradi R, Abbasian Arani AA. Multi-objective optimization of cost and thermal performance of double walled carbon nanotubes/water nanofluids by NSGA-II using response surface method. *Applied Thermal Engineering*. 2017;112: 1648–1657. https://doi.org/10.1016/j.applthermaleng.2016.10.129
33. Abdollahi A, Shams M. Optimization of heat transfer enhancement of nanofluid in a channel with winglet vortex generator. *Applied Thermal Engineering.* 2015;91: 1116–1126. https://doi.org/10.1016/j.applthermaleng.2015.08.066
34. Arora A, Bajaj I, Iyer SS, Hasan MMF. Optimal synthesis of periodic sorption enhanced reaction processes with application to hydrogen production. *Computers & Chemical Engineering*. (2018);*115:* 89–111.https://doi.org/10.1016/j.compchemeng.2018.04.004
35. Iyer SS, Bajaj I, Balasubramanian P, Hasan MMF. Integrated Carbon Capture and Conversion To Produce Syngas: Novel Process Design, Intensification, and Optimization. *Industrial & Engineering Chemistry Research.* (2017);*56*(30): 8622–8648.
36. https://doi.org/10.1021/acs.iecr.7b01688
37. Liu J, Ploskas N, Sahinidis NV. Tuning BARON using derivative-free optimization algorithms. *Journal of Global Optimization*. 2019;*74*(4): 611–637. https://doi.org/10.1007/s10898-018-0640-3

Lukas Peters: https://orcid.org/0009-0005-8195-8177

Rüdiger Kutzner: https://orcid.org/0009-0005-7171-024X

Marc Schäfer: https://orcid.org/0009-0009-2265-7023

Lutz Hofmann: https://orcid.org/0000-0002-3688-6136