

## Article

# On the Effectiveness of Optimisation Algorithms for Hydrodynamic Lubrication Problems

František Kocman  and Pavel Novotný 

Institute of Automotive Engineering, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2896/2, 616 69 Brno, Czech Republic; frantisek.kocman@vut.cz

\* Correspondence: novotny.pa@fme.vutbr.cz

**Abstract:** In many applications, it is necessary to optimise the performance of hydrodynamic (HD) bearings. Many studies have proposed different strategies, but there remains a lack of conclusive research on the suitability of various optimisation methods. This study evaluates the most commonly used algorithms, including the genetic (GA), particle swarm (PSWM), pattern search (PSCH) and surrogate (SURG) algorithms. The effectiveness of each algorithm in finding the global minimum is analysed, with attention to the parameter settings of each algorithm. The algorithms are assessed on HD journal and thrust bearings, using analytical and numerical solutions for friction moment, bearing load-carrying capacity and outlet lubricant flow rate under multiple operating conditions. The results indicate that the PSCH algorithm was the most efficient in all cases, excelling in both finding the global minimum and speed. While the PSWM algorithm also reliably found the global minimum, it exhibited lower speed in the defined problems. In contrast, genetic algorithms and the surrogate algorithm demonstrated significantly lower efficiency in the tested problems. Although the PSCH algorithm proved to be the most efficient, the PSWM algorithm is recommended as the best default choice due to its ease of use and minimal sensitivity to parameter settings.

**Keywords:** journal bearing; thrust bearing; hydrodynamic lubrication; particle swarm algorithm; pattern search; surrogate algorithm; genetic algorithm



Received: 24 March 2025

Revised: 5 May 2025

Accepted: 6 May 2025

Published: 8 May 2025

**Citation:** Kocman, F.; Novotný, P. On the Effectiveness of Optimisation Algorithms for Hydrodynamic Lubrication Problems. *Lubricants* **2025**, *13*, 207. <https://doi.org/10.3390/lubricants13050207>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Many machines use hydrodynamic (HD) bearings to regulate the interaction between a moving rotor and a stator. HD journal and thrust bearings are common examples. Their advantages and disadvantages are well established. A notable disadvantage is friction losses caused by shear stresses in the thin lubrication layer, which are frequently evaluated alongside load-carrying capacity (LCC) in various applications.

HD lubrication involves the flow of a viscous fluid coupled with heat transfer and interaction with elastic bodies. The continuity equation, Navier–Stokes equation, energy equation and equations describing the elastic deformation of bodies govern HD lubrication. Historically, the approach has been to simplify the problem by neglecting certain effects, leading to the derivation of the Reynolds equation. The Reynolds equation typically describes a two-dimensional (2D) problem and can be solved using numerical methods. Under significant constraints, the Reynolds equation can also be solved analytically using various approaches; for example, short bearing theory is commonly applied for specific types of bearings [1]. However, for some specific bearings, the significant simplifications assumed in the Reynolds equation are no longer valid. In such cases, it is necessary to

adopt an approach that considers general three-dimensional (3D) flow, typically solved numerically through computational fluid dynamics (CFD) [2–5].

A typical challenge in designing an HD journal bearing is selecting its design parameters, such as journal diameter, journal width, bearing clearance and other factors depending on the complexity of the design. This is followed by analysing the effect of these design parameters on the bearing's properties to ensure sufficient LCC, minimise frictional losses or reduce lubricant flow rates. Optimising bearing performance can be conducted for a single operating condition or an entire range of operating conditions, typically using computational modelling at various physical levels [6,7].

The level of physical detail in the description of HD lubrication, which influences the type of solution—empirical, analytical or numerical—directly affects the time required to solve the problem. While empirical and analytical solutions can provide results for a single operating condition almost instantly, detailed 3D models may take hours to compute. The time required to solve a problem significantly impacts both the approach to designing an HD bearing and the strategy chosen to achieve optimal performance.

The number of variable parameters in an HD bearing also significantly impacts the choice of optimisation strategy. In the case of a small number of parameters, typically one or two, the problem is relatively easy to solve, for example, by parametric studies. However, with a larger number of variable parameters, it becomes quite difficult to design an optimal bearing using this approach, and an optimisation algorithm must be used. In cases where 3D models are applied, it is absolutely crucial to choose an optimisation algorithm that minimises the number of partial solutions, i.e., the minimum number of evaluations of the objective function.

Thus, the aim of this work is to identify an efficient optimisation algorithm capable of finding the HD bearing with optimal performance while minimising the number of objective function evaluations. Given that different optimisation algorithms involve numerous variations in settings and input parameter values, this work also aims to analyse in detail the effect of these algorithm settings. The methods used to obtain the HD bearing lubrication solution and the objective function are not the primary focus of this study.

## 2. Review of the Current State of the Art

HD lubrication and related computational methods for the design, analysis and optimisation of bearings have been extensively published in the world literature, as shown in [8–12]. Research on optimisation algorithms has evolved in a similar dynamic manner to research on computational methods for describing HD lubrication. For example, Nicoletti [8] presented a local optimisation algorithm approach using a 2D numerical solution of the Reynolds equation to express the objective function. In this approach, the radius was chosen as the optimisation parameter, described by a cubic spline as a function of the angle, with the objective of achieving a higher limit of rotor stability. The optimisation algorithm employed was a gradient method, specifically sequential quadratic programming (SQP). However, as the author himself pointed out, this method does not guarantee finding the global minimum.

In contrast, Ramos and Daniel [9] used a more complex lubrication model, coupled with the energy equation, and employed the finite volume method for discretising the Reynolds equation. The goal of this optimisation was to increase the bearing's LCC, reduce the viscous shear force and minimise the heat generated in the lubrication layer. The dimensions of the micro-grooves on the inner surface of the bearing were chosen as optimisation parameters. Unlike Nicoletti [8], the gradient method was combined with a global optimisation algorithm, specifically particle swarm (PSWM) optimisation. This combination was selected for solution stability, with PSWM used to

find the global minimum (coarse solution), followed by refinement using the gradient-based optimisation algorithm. Although the authors outlined the steps for selecting a suitable algorithm, they did not provide a specific rationale for choosing the particular global optimisation algorithm.

A similar approach was used by Hashimoto and Matsumoto [10]. The objective function included the outlet oil flow rate, whirl onset velocity of the journal and maximum averaged oil film temperature rise, all of which were given equal weight. Radial clearances, bearing length-to-diameter ratio and bearing orientation angle were chosen as optimisation parameters for the elliptical bearing. The Reynolds equation, modified to account for the effect of turbulent flow in the lubrication layer, was solved using the finite element method. Optimisation was first performed using the direct search method to find suitable initial values for the optimised parameters, followed by the use of SQP to find the minimum.

The opposite approach is presented by Zhang et al. [11]. In this case, SQP is integrated into a multi-objective genetic algorithm (GA), specifically used to calculate relative eccentricity. This model is further enhanced with an inversion verification process, which improves the stability and accuracy of the solution. Unlike the previous approaches, three objective functions are calculated: power loss, oil leakage and friction torque.

Many optimisation strategies have also been developed to optimise the performance of HD thrust bearings. A relatively common approach is the use of more computationally intensive models for multiphysics lubrication analysis, namely CFD. Ostayen [12] demonstrated an approach using the so-called one-shot optimisation procedure, where Lagrange multipliers are used to solve the Reynolds equation. The resulting solution satisfied the specified conditions, namely the maximum lubrication layer thickness, while ensuring sufficient bearing LCC at a given velocity. Unlike previous authors, Ostayen also presented a time-domain solution, allowing the optimal solution to be achieved at different speeds and loads. However, as the author himself noted, this approach is primarily suitable for 1D and 2D thrust bearing solutions, mainly due to time constraints.

A large number of strategies are based on gradient-based methods. For example, Cheng and Chang [13] used a conjugate gradient method in conjunction with a direct HD lubrication solver. An optimisation method using the gradient method was also presented by Rajan et al. [14]. Fesanghary et al. [15] applied the SQP method to optimise the sectorial thrust bearing.

An integrated design and optimisation philosophy for the entire air bearing rotor system using GA was presented by Saruhan et al. [16], along with a comparison of GA optimisation with traditional methods in their work [17]. These approaches are typically suitable for the general design of the bearing lubrication gap and allow for the consideration of many design parameters.

An ongoing effort by researchers is to increase the physical depth of computational models to describe the problem in more detail. Optimisation approaches using computationally intensive models that describe the HD thrust bearing lubrication problem in greater detail were presented, for example, by Charitopoulos et al. [4]. The lubrication model used a CFD-based approach to analyse taper-land-type and pocket-type thrust bearings. The authors investigated the influence of various parameters on both bearing designs, with optimisations performed using GA. For the taper-land-type bearing, two parameters were selected for optimisation: the maximum taper part height and the ratio between the taper part and land part area of the thrust bearing. The result was a reduction of more than 7% in power loss at a rotor speed of 200,000 rpm compared to the original design. Similarly, lubrication was optimised using GA and a CFD model in the work of Fouflias et al. [2] for curved-pocket thrust bearings, aiming to maximise bearing load capacity and minimise frictional losses. The results showed a significant improvement compared to other relevant

studies, with load capacity increased by up to 16% and the friction coefficient reduced by 21%. The use of GA in the optimisation of micro-thrust bearings was also presented by Papadopoulos et al. [18].

From the above literature, it is clear that the most commonly used optimisation algorithms are SQP and GA. Two conclusions can be drawn. The first is that authors often fail to provide reasons for choosing a particular optimisation algorithm or to present the criteria for selecting it. This may be due to a lack of knowledge about suitable algorithms for the problem or the limitations of the commercial tools available. The second conclusion, which is quite expected, is that the computational complexity of HD lubrication solutions continues to increase. Despite the rise in computing power, solution times are not decreasing. As a result, the number of bearing variants considered for optimisation may be limited, the number of optimisation input parameters may be reduced, or optimisation may be carried out, for only one or a small fraction of the operating conditions.

This work focuses on selecting an efficient optimisation algorithm for determining the optimal parameters of HD bearings under specified operating conditions. Two types of HD bearings were selected: an oil-film journal bearing and a segmented, double-sided oil-film thrust bearing, both of which are used in an industrial turbocharger. Various optimisation algorithms, along with their sub-algorithms and different parameter settings, were analysed for these two bearings.

Finding an efficient optimisation method is divided into two parts. The first part involves analysing the appropriate parameter settings for a given optimisation algorithm. The second part focuses on analysing the individual optimisation algorithms, with the definition of the objective function as outlined by Novotný et al. [19].

### 3. Strategy for Finding an Efficient Algorithm

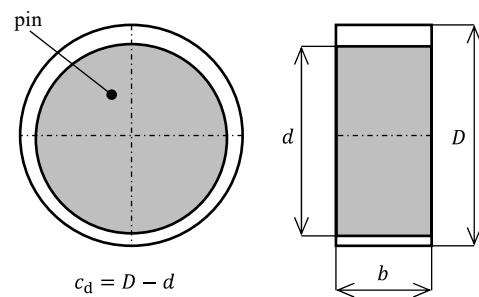
The choice of optimisation algorithm depends, to some extent, on the physical nature of the problem. The optimisation algorithms were tested on both the analytical and numerical solutions of an HD journal bearing and the numerical solution of an HD thrust bearing. It can be assumed that both the journal bearing and the thrust bearing exhibit similar characteristics in the objective function space.

The best settings for specific parameters of the optimisation algorithm, or partial variants of the algorithm, were determined only using the analytical solution of the journal bearing that is presented in Appendix A.1. A typical analytical calculation of the objective function, which describes the HD bearing properties, is extremely fast and allows for the comparison of many parameter setting variants.

Due to computational complexity, numerical models are used only to investigate the behaviour of individual optimisation algorithms with a pre-selected optimal setting. This approach is applied to both the HD journal bearings and the HD thrust bearings and the results are presented in Section 4.3. As previous studies [9,19] have shown that a more complex description of the lubrication layer results in a greater number of local minima, only global algorithms, or a combination of global and local algorithms, were chosen.

#### 3.1. Definition of Bearing Design Parameters and Operating Conditions

The geometric dimensions of the HD journal bearing, used for searching the optimal settings of the optimisation algorithm, are presented in Figure 1. The values of the geometric dimensions, along with the range of possible values, are provided in Table 1.

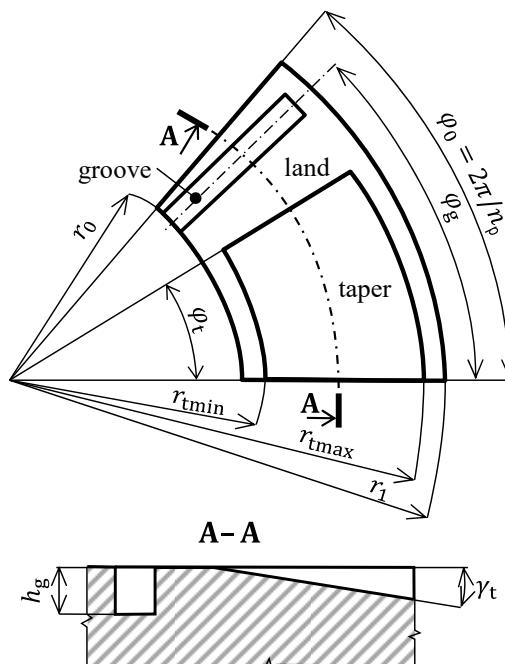


**Figure 1.** Definition of the dimensions of a HD journal bearing, where  $d$  is the journal diameter,  $D$  is the shell diameter,  $b$  is the journal width and  $c_d$  is the bearing clearance.

**Table 1.** Journal bearing design parameters.

Parameter	Value	Range
Bearing diameter, $d$ [mm]	32	28–32
Bearing width, $b$ [mm]	17	13–17
Bearing clearance, $c_d$ [mm]	0.050	0.020–0.070

The selected thrust bearing is typically used in turbochargers and is designed to carry axial forces acting in both directions: the compressor pulling direction (positive values) and the turbine pulling direction (negative values). The bearing must therefore be designed as double-sided, containing two lubrication gaps oriented perpendicular to the axis of rotation. The lubrication gap of the thrust bearing is formed by a working surface on the bearing disc and a planar surface on the thrust ring. The definition of the geometric dimensions of one segment of the working surface is shown in Figure 2. This definition applies to both segments on the thrust side and the counter-thrust side of the bearing.



**Figure 2.** Definition of the dimensions of the working surface of one thrust bearing segment. The symbol  $n_p$  represents the number of segments,  $r_0$  is the inner radius of the working surface,  $r_1$  is the outer radius of the working surface,  $r_{t\min}$  is the inner radius of the tapered part,  $r_{t\max}$  is the outer radius of the tapered part,  $\gamma_t$  is the taper wedge angle,  $\gamma_{t\max}$  is the limit wedge angle of the tapered part,  $h_g$  is the groove height,  $\varphi_t$  is the taper part angle, and  $\varphi_g$  is the angle of the lubricating groove.

The values of the design parameters for the thrust bearing, as shown in Figure 2, along with the possible ranges for each parameter, are given in Table 2. These values apply to all segments on both the thrust and anti-thrust sides.

**Table 2.** Thrust bearing design parameters. The values are valid for both the thrust and counter-thrust sides of the bearing.

Parameter	Value	Range
Number of pads, $n_p$ [-]	12	—
Axial clearance, $c_{ax}$ [mm]	0.2	—
Taper angle, $\varphi_t$ [ $^\circ$ ]	26	10–30
Wedge taper angle, $\gamma_t$ [ $^\circ$ ]	0.2	0.1–1
Inner radius, $r_0$ [mm]	75	—
Outer radius, $r_1$ [mm]	130	120–140
Taper inner radius, $r_{tmin}$ [mm]	77	—
Taper outer radius, $r_{tmax}$ [mm]	128	120–140
Groove angular position, $\varphi_g$ [ $^\circ$ ]	0	20–30
Groove height, $h_g$ [mm]	0.5	—

The performance of the journal and thrust bearings is considered under the operating conditions defined in Table 3. These conditions correspond to a medium-sized turbocharger for a stationary internal combustion engine. The values for bearing load capacity limits and lubricant flow rates through the bearing are determined for the reference bearing based on the operating conditions. More detailed information can be found in [19].

**Table 3.** Hydrodynamic bearing operating conditions.

Parameter	Value	1	2	3	4
Operating condition no., [-]					
Rotor speed, $n$ [rpm]	5000	15,000	20,000	30,000	
Relative eccentricity, $\varepsilon$ [-]	0.32	0.24	0.3	0.21	
Radial pin velocity, $v_R$ [ $m \cdot s^{-1}$ ]	0	0	0	0	
Oil temperature at inlet, $T_{in}$ [ $^\circ$ C]	60	60	60	60	
Dynamic viscosity, $\mu$ [ $Pa \cdot s$ ]	0.005	0.005	0.005	0.005	
Oil density $\rho$ [ $kg \cdot m^{-3}$ ]	840	840	840	840	
Oil heat capacity $C$ [ $J \cdot m^{-1} K^{-1}$ ]	2200	2200	2200	2200	
Load capacity limit of journal bearing $F_{lim}$ [N]	100	100	150	150	
Mass flow rate limit of journal bearing $\dot{m}_{lim}$ [ $kg \cdot s^{-1}$ ]	0.05	0.05	0.05	0.05	
Load capacity limit of thrust bearing $F_{lim}$ [N]	250	480	670	1068	
Mass flow rate limit of thrust bearing $\dot{m}_{lim}$ [ $kg \cdot s^{-1}$ ]	0.11	0.16	0.20	0.30	

Table 3 combines the operating conditions for both the journal and thrust bearings and includes typical properties describing the kinematic quantities of the journal relative to the shell, as well as the lubricant properties. The relative eccentricity for a journal bearing is defined as  $\varepsilon = 2e/c_d$ , where  $e$  is the pin eccentricity. For the thrust bearing, the relative eccentricity is defined as follows:

$$\varepsilon = 1 - \frac{h_{0ts}}{c_{ax}}, \quad (1)$$

where  $h_{0ts}$  is the minimum thickness of the lubrication gap on the thrust side of the bearing.

### 3.2. Definition of the Objective Function

The aim of the optimisation is to find the bearing geometric parameters that minimise the friction moment  $M_f$  while satisfying two conditions:

The LCC of the bearing must not be significantly lower than the limit force ( $F_{lim}$ ).

The lubricant flow rate through the bearing must not be significantly higher than the flow rate limit ( $\dot{m}_{lim}$ ).

The terms ‘significantly higher’ or ‘significantly lower’ are defined by a smooth transition of a given quantity from an acceptable level to an unacceptable level, which is expressed using a continuous exponential function in Equations (5) and (6).

The vector of design parameters for optimisation ( $\mathbf{x}$ ) that forms the optimisation space for the journal bearing case is defined as

$$\mathbf{x} = [d, b, c_d]. \quad (2)$$

The vector of design parameters for the optimisation of the thrust bearing is defined as

$$\mathbf{x} = [\varphi_{t,ts}, \gamma_{t,ts}, r_{1,ts}, r_{tmax}, \varphi_g]. \quad (3)$$

The possible values of these parameters were constrained by the values given in Tables 1 and 2. The objective function ( $f_{obj}$ ) used to minimise the friction torque while constraining the load capacity and lubricant flow rate is defined by the following equation:

$$f_{obj} = s_M s_F s_{\dot{m}}, \quad (4)$$

where  $s_M$  is the friction torque ratio,  $s_F$  is the load capacity correction factor, and  $s_{\dot{m}}$  is the flow rate correction factor. This strategy uses the so-called reference bearing, which serves as the default state and typically represents the bearing used in serial production. However, any other bearing parameters may be used for the base bearing. Based on this strategy, the friction torque ratio is defined as

$$s_M = \frac{M_f}{M_{f,ref}}, \quad (5)$$

where  $M_f$  is the current friction torque, and  $M_{f,ref}$  is the friction torque of the reference bearing. In this strategy, two correction factors,  $s_F$  and  $s_{\dot{m}}$ , are defined as ratios using actual and limit values, as follows:

$$s_F = \max \left( 1, \left( \frac{F_{lim}}{F_c} \right)^3 \right), \quad (6)$$

$$s_{\dot{m}} = \max \left( 1, \left( \frac{\dot{m}_o}{\dot{m}_{lim}} \right)^3 \right). \quad (7)$$

If the values of the LCC  $F_c$  or mass flow rate  $\dot{m}_o$  are within the predefined limits  $F_{lim}$  and  $\dot{m}_{lim}$ , the objective function value remains unchanged. However, if the values of  $F_c$  or  $\dot{m}_o$  are outside the limits, the objective function value rises sharply and continuously due to the cubic exponent.

The limit values allow other requirements for the HD bearing to be met. Typically, it is necessary to maintain a load capacity at a given operating point that corresponds to at least the proportion of the weight load in the bearing or to limit unwanted increases in lubrication system requirements. In the case of this work, the limit values based on the serial bearing used in a given turbocharger are provided in Table 3. Furthermore, the value  $M_{f,ref} = 0.1$  Nm is used for the analytical solution of the journal bearing.

### 3.3. Optimisation Algorithms

The algorithms chosen to compare the efficiency of the optimisation algorithms were those commonly used in HD bearing optimisation [9,10,20–22], namely PSWM, GA, pattern search (PSCH) and surrogate (SURG). In this section, only the short bearing model presented in Appendix A.1 was used. The following sections describe the optimisation algorithms used, the sub-variants of these algorithms tested and the parameter settings of these algorithms.

#### 3.3.1. Particle Swarm Algorithm

It is an evolutionary optimisation algorithm based on the movement of particles (swarm) in a user-defined space. Each particle is defined by its position, velocity and memory of previous search successes. The particles are initially randomly distributed (initial position) and move in random directions with a defined velocity. The particles are influenced by the more successful particles of the swarm. The algorithm computes the movement of the swarm in discrete time steps and continuously adjusts the values describing the particles. The optimisation gradually leads to the convergence of the particles towards a region of the global minimum.

The version of the algorithm used is based on the one presented by Kennedy and Eberhart [22], with modifications by Mezura-Montes and Coello Coello [23] and Pedersen and Chipperfield [24]. The initial random distribution of particles is directly influenced by the initial swarm span parameter, with particles generated over an interval of  $\pm$  initial swarm span, scaled by the boundary conditions.

The size of the neighbourhood affecting other particles is determined by the minimum neighbours fraction parameter, which represents the fraction of points from the total swarm. Since this parameter directly influences the number of objective function evaluations, five values were selected for the algorithm analysis, considering both extremes: zero influence of surrounding particles and influence of the entire swarm on each other.

Three weighting factors are used to calculate the new speed: the inertia range, Self adjustment weight and Social adjustment weight. The first factor defines the upper and lower inertia limits, which serve as the weighting factor for the original speed. The second weighting factor determines the significance of the best position of a given particle. The final factor defines the weighting of the nearby neighbourhood. Since the chosen version of the algorithm uses a variable inertia value, its limiting values were selected based on [24–26].

An important decision was the size of the swarm. As Pedersen and Chipperfield [24] point out, the commonly recommended values, such as those given in [25], are not universally applicable. This issue was discussed in more detail by Piotrowski et al. [26], who suggested that in practical applications, it is preferable to choose values between 70 and 500. Given the larger number of setting variations already selected and the fact that the analysis was performed on a significantly simpler model, a swarm size of 100 was chosen.

Similarly, as stated by Ramos and Daniel [9], variants in which the PSWM is augmented with a local algorithm using gradient methods [27–30] and a global PSCH algorithm [31] have also been analysed.

The optimisation termination was determined using two parameters: the maximum number of iterations (Max. Iterations) and the maximum number of iterations without a change in the objective function of the best individual, when the difference between two consecutive values is less than the function tolerance (Max. Stall Iterations).

Table 4 lists all the above settings, their tested values and their settings in the case of PSWM.

**Table 4.** Algorithm settings used for PSWM.

Option	List of Tested Values and Options
Initial swarm span	2000; 1000; 500; 3000; 5000; 10,000
Minimum neighbours fraction	0.25; 0; 0.5; 0.75 1
Inertia range	0.1–1.1; 0.1–2.1; 0.1–3.1
Self adjustment weight	1.49; 1.25; 1.75; 2
Social adjustment weight	1.49; 1.25; 1.75; 2
Swarm size	100
Hybrid function	none; SQP [27–30]; Pattern search [31]
Function tolerance	$10^{-6}$
Max. Iterations	600
Max. Stall Iterations	20

### 3.3.2. Genetic Algorithm

The GA is a heuristic optimisation algorithm, belonging to the class of evolutionary algorithms, inspired by the natural selection process in evolutionary biology. The optimisation process begins with the formation of an initial population of individuals (solutions), where each individual is characterised by genes (parameters) distributed within a defined optimisation space. This space is typically constrained by boundary conditions, such as upper and lower bounds. The population size significantly impacts both the convergence rate and the overall computational complexity of the algorithm [32]. While a larger population provides a more detailed search of the space, it also increases computational complexity. Therefore, the population size range was selected based on the recommendations in [32,33].

Since simple selection based on the value of the objective function could lead to a dead end, the scores of individuals need to be scaled, a process known as fitness scaling. Four fitness scaling functions were tested. Rank fitness scaling evaluates individuals based on the value of the objective function, with the best individual receiving a score of 1 and the last one close to 0 [34]. A simpler method, top fitness scaling, assigns a scaling score equal to the original score times the population size to the top 40% of individuals, while the rest receive a score of 0 [34]. In contrast, the more sophisticated linear shift fitness scaling ensures that the expected best individual has a score equal to twice the average score. This results in a linear shift of all individuals' scores [34]. Finally, proportional fitness scaling normalises only the original score, with its suitability strongly depending on the composition of the population.

From this initial population, the next generation is created by selecting elite individuals (elite children), crossing over two individuals (crossover children) or mutating genes (mutation children). The selection of elite individuals to move on to the next generation depends on two parameters: their current score and the percentage of the best individuals (elite fraction) that will advance. The frequency of elite individuals has a significant impact on finding the global minimum [35].

The aim of crossover is to produce two new individuals from each pair for the next generation. Six crossover functions were tested. The proportion of individuals (parents) that will be used in the next generation to create new individuals (offspring) by crossover is given by the parameter crossover fraction. If this fraction is high, large intergenerational differences will occur, which affect convergence. Therefore, this parameter was chosen in the recommended range of 0.4–0.85 [36]. The method by which individual offspring are formed is one of the analysed parameters. Laplace crossover [37] uses the relations  $c_{j,1} = p_{j,1} + \beta_j |a_{j,1} - a_{j,2}|$  and  $c_{j,2} = p_{j,2} + \beta_j |p_{j,1} - p_{j,2}|$ , where  $a_{j,1}$  and  $a_{j,2}$  are the parents,  $c_{j,1}$  and  $c_{j,2}$  are the children and  $\beta_j$  is a random number generated from the Laplace distribution, to form the children. A more complex method is heuristic crossover, where the chromosomes passed on depend on the distance from the parents [38,39]. In contrast,

a relatively straightforward method is One-point crossover [40], where a random value of  $n$  is generated between 1 and the number of variables. Subsequently, an offspring is created that has chromosomes from 1 to  $n$  from the first parent and chromosomes from  $n$  to the number of variables from the second parent. Similarly, Two-point crossover works by including a second random point [40]. In the case of Arithmetic crossover, the children are the weighted average of both parents [41]. Scattered crossover [40] generates a binary vector; if the value of a given chromosome is equal to 1, it is passed from the first parent; otherwise, it is passed from the second parent.

Another way of creating offspring is through mutation, where one or more genes are changed based on the applied function [39]. Four mutation functions were tested. Gaussian mutation changes individual genes by adding a random number from a Gaussian distribution [42]. In contrast, ‘mutationadaptfeasible’ is similar in principle to the generalised pattern search (GPS) [31], where the mutation of an individual depends on its position in the optimisation space. Power mutation is inspired by the power distribution, where the form of the resulting offspring depends on the scaled distance of the parent from the boundaries of the optimised parameter and the position of the current parent [39]. The last mutation function tested is positive basis mutation, which is, more precisely, orthogonal mesh adaptive direct search (MADS) [43].

In addition to the scaled score, the selection function also influences which individual will be selected as a parent for the next generation. Five selection functions were tested. The recommended Tournament selection selects parents through a tournament among several chosen individuals, with the best of them becoming the parents [44]. In contrast, Remainder selection [44] works in two steps. First, a position is assigned to a parent based on the parent’s scaled score, specifically the integer part. In the second step, a second function, Selection roulette, is applied [44]. Each parent is represented on an imaginary roulette wheel and then randomly selected. The next function tested is Uniform selection, where selection depends only on the number of parents and their probability of success [44]. Stochastic universal selection [44] initially lines up each parent in turn to form a line, with the length of each parent’s segment proportional to its scaled score. It then moves along this line with a constant random step, selecting individual parents.

Similar to the PSWM, variants in which this algorithm is complemented by a global PSCH algorithm [31] and a local algorithm based on the gradient method were analysed [27–30]. Three conditions were used to avoid unnecessary computation time.

The optimisation was terminated when the difference between two successive values of the objective function was less than or equal to the Function tolerance, and when the Max. Stall Generations were reached. Additionally, parameters such as the maximum number of generations (Max. Generations) and the maximum computation time in seconds (Max. Stall Time) were used to limit excessive solution times.

Table 5 lists all the above settings, their tested values and their settings in the case of GA.

**Table 5.** Algorithm settings used for GA.

Option	List of Tested Values and Options
Population size	50; 100; 200; 500; 1000
Fitness scaling	Rank fitness scaling [34]; top fitness scaling [34]; linear shift fitness scaling [34]; proportional fitness scaling;
Elite fraction	0.05
Crossover fraction	0.7

**Table 5.** Cont.

Option	List of Tested Values and Options
Crossover function	Laplace crossover [37]; heuristic crossover [38,39]; One-point crossover [40]; Two-point crossover [40]; Arithmetic crossover [41]; Scattered crossover [40]
Mutation function	Gaussian mutation [42]; mutationadaptfeasible [31]; Power mutation [39]; positive basis mutation [43]
Selection function	Tournament selection [44]; Remainder selection [44]; Roulette selection [44]; Uniform selection [44]; Stochastic universal selection [44]
Hybrid function	none; Pattern search [31]; SQP [27–30]
Max. Generations	80
Max. Stall Generations	10
Function tolerance	$10^{-6}$
Max. Stall Time	300

### 3.3.3. Pattern Search Algorithm

The PSCH algorithm, sometimes referred to as the direct search algorithm, belongs to the family of direct search algorithms used for optimising various functions. Unlike previous algorithms, PSCH includes several sub-algorithms, and its execution process can be divided into two phases: the search and the poll phases. For the purpose of describing PSCH, its basic version, GPS [31], was selected. In the first phase, the objective function is evaluated on a predefined grid (search), and when a better position is found, the grid is shifted to that position. If no better position is found, the algorithm proceeds to the second phase, where the network is updated (depending on the setting and type of the algorithm), and the objective function is evaluated at the new positions (poll).

The main influence on the settings comes from the choice of the specific algorithm. For example, the so-called classic algorithm allows for specific settings for both the search and poll phases. Without explicit specifications, this corresponds to the GPSPositiveBasis2N algorithm (for both search and poll) [31], which can also utilise mesh rotation.

Two versions of the patterns were used for the given poll or search methods. In the case of the PositiveBasis2N label, the pattern for the three optimised parameters is as follows:  $[1\ 0\ 0][0\ 1\ 0][0\ 0\ 1][-1\ 0\ 0][0\ -1\ 0][0\ 0\ -1]$ . For the so-called PositiveBasisNp1, the pattern is:  $[1\ 0\ 0][0\ 1\ 0][0\ 0\ 1][-1\ -1\ -1]$ . Six poll methods were tested, more specifically three methods, but for both patterns. In addition to GPS [31,45], Generating Set Search (GSS) [46–48] and MADS [43,49] were also used. These poll methods were also employed as search functions.

However, other algorithms, such as the genetic algorithm (searchGA) [50], Latin hypercube search (searchLHS) [51], Nelder–Mead algorithm (searchNelderMead) [52] and radial basis function surrogate (RBFsurrogate) [53–55] can also be used for the search. Thus, 10 search functions were tested.

Since there may be situations where a poll occurs for a location that has been previously evaluated, a setting has been included to remember these positions (Cache). It can also be configured whether all new points on the grid are evaluated or if the next step occurs only when the position of the new point is better than the existing one (Use complete poll). A similar setting applies to the search (Use complete search). The poll rate can also be influenced by the way the points are evaluated, namely whether they are evaluated in the order they were generated, randomly, or by selecting a point that has the same direction as the previous successful point.

In the case where the Nonuniform Pattern Search (NUPS) algorithm is used, which is a specific combination of the previously mentioned poll methods, the poll and grid settings cannot be set explicitly.

As with the previous algorithms, unnecessary prolongation of the calculation needed to be avoided. To achieve this, five conditions were used: the maximum number of iterations (Max. Iterations), the maximum number of evaluations of the objective function (Max. Function Evaluations), the difference in the positions of two consecutive iterations (Function tolerance), the change in the mesh size (Step Tolerance) and the minimum mesh size (Mesh Tolerance).

Table 6 lists all the above settings, their tested values and their settings in the case of PSCH.

**Table 6.** Algorithm settings used for PSCH.

Option	List of Tested Values and Options
Algorithm	classic; NUPS [31,43,45,49]; NUPS-GPS [46–48]; NUPS-MADS [43]
Cache	on; off
Mesh rotate	on; off
Poll method	GPSPositiveBasis2 [31]; GPSPositiveBasisNp1 [31]; GSSPositiveBasis2N [46]; GSSPositiveBasisNp1 [46]; MADSPositiveBasis2N [49]; MADSPositiveBasisNp1 [49]
Search Function	GPSPositiveBasis2N [31]; GPSPositiveBasisNp1 [31]; GSSPositiveBasis2N [46]; GSSPositiveBasisNp1 [46]; MADSPositiveBasis2N [49]; MADSPositiveBasisNp1 [49]; searchGA [50]; searchLHS [51]; searchNelderMead [52]; RBFsurrogate [53–55]
Use complete poll	true; false
Use complete search	true; false
Poll order algorithm	Consecutive; Random; Success
Max. Iterations	300
Max. Function Evaluations	6000
Function Tolerance	$10^{-6}$
Step tolerance	$10^{-6}$
Mesh Tolerance	$10^{-6}$

### 3.3.4. Surrogate Algorithm

This algorithm is based on a surrogate function. In principle, it can be divided into two phases.

In the first phase, a surrogate function is constructed using a radial basis function [53–55] over random points that have been evaluated by the initial objective function. The number of these initial points (Minimum surrogate points) significantly affects the accuracy of the surrogate function and the time required to construct it [53].

In the second phase, a large number of points near the current minimum of the objective function are generated. These points are evaluated using the surrogate function, and the resulting values are then used as input to the merit function. A parameter, including the distance of these points from the points evaluated by the objective function, also contributes to the merit function. The point that minimises the merit function is then

evaluated by the objective function, and this result is used to update the surrogate function. The number of points or the timing of the surrogate function update is explicitly defined by the Batch update interval parameter. A larger number of points results in higher accuracy but also increases computational complexity [53].

The maximum number of function evaluations (Maximum of function evaluations) was used as a termination condition.

Table 7 lists all the above settings, their tested values and their settings in the case of SURG.

**Table 7.** Algorithm settings used for SURG.

Option	List of Tested Values and Options
Minimum surrogate points	10; 20; 30; 50; 70
Batch update interval	1; 5; 10; 15; 20
Maximum of function evaluations	200; 500; 1000; 1500

### 3.4. Methodology for Evaluating the Impact of Algorithm Parameter Settings on Its Efficiency

A key requirement for a global optimisation algorithm is its ability to find the global extreme. However, in practical applications, the time required to achieve this, i.e., finding the global minimum value of the objective function ( $f_{\text{obj}}$ ), plays a significant role. Thus, the evaluation of the efficiency of optimisation algorithms was based on their ability to find the global extreme and the speed of achieving it.

The speed of the optimisation computation depends not only on the efficiency of the algorithm itself, including the number of iterations or generations, but also on the performance of the hardware. To eliminate the influence of the hardware on which the computations were performed, the number of evaluations of the objective function ( $n_f$ ) was chosen as the metric for speed evaluation.

Since various modifications and parameter settings of individual optimisation algorithms were analysed, it can be assumed that they have differing levels of influence on both computational demand and the value of the objective function. Therefore, it is essential to isolate the effects of settings other than those currently being evaluated. For this purpose, the methodology presented by McGill et al. [56], with modifications by Cox [57], was employed. The objective function values for each variant of the algorithm settings are represented using five key metrics: the whiskers (1.5 times the interquartile range), the upper and lower hinges (quartiles) and the median. Values that lie above or below the whiskers, referred to as outliers, are presumed to result from the significant influence of parameters other than the one being evaluated. This approach eliminates the need to verify the type of distribution in the analysed data set, as the methodology is non-parametric [57].

The optimal settings for a given optimisation algorithm were determined exclusively for the analytical expression of the objective function in the journal bearing case. Individual optimisation algorithms were analysed with numerous settings. However, the total number of combinations for a given algorithm's settings is constrained compared to the theoretical number of combinations due to the incompatibility of certain settings. To minimise the influence of pseudo-random factors, such as the selection of initial parameters that depend on the random number generator, 10 optimisation iterations were performed for each algorithm setting. If a particular setting resulted in a violation of the input parameter value interval three consecutive times, the solution was aborted. The number of settings analysed for each algorithm is shown in Table 8. These values reflect the count of unique variants, excluding the number of iterations for a given setting. All tasks were executed on a single core (without parallelisation).

**Table 8.** Overview of the analysed number of settings for each algorithm without repetition to limit the initial seed of individuals.

Function	Number of Variants
GA	26,100
PSWM	4320
PSCH	11,328
SURG	100
Total	41,848

The individual optimisation methods, using the best algorithm settings, were then applied to optimise the journal and thrust bearing performances with a numerical formulation of the objective function. The effectiveness of each algorithm was evaluated based on its ability to find the global extremum and the number of evaluations of the objective function.

## 4. Results and Discussion

### 4.1. Evaluation of the Influence of Settings and Parameter Values of Optimisation Algorithms

To evaluate the effect of the optimisation algorithm's settings and parameter values, the minimum of the objective function was manually determined, emphasising compliance with boundary and operating conditions. Subsequently, the individual algorithms were tested using variations in algorithm settings and parameter values on examples of analytical journal bearing solutions. The results are summarised in Table 9.

**Table 9.** Efficiency evaluation for journal bearing optimisation using the analytical solution of the objective function for all algorithms and settings.

Function	-	PSWM	GA	PSCH	SURG
Number of objective function evaluations, $n_f [-]$	Min	746	570	20	195
	Max	5160	77,554	3510	1500
	Mean	1771	15,063	517	786
	Std	470.47	16,849.00	507.85	493.97
Objective function value, $f_{obj} [-]$	Min	0.7011	0.0098	0.7011	0.7199
	Max	0.7026	0.9823	1.0766	0.7618
	Mean	0.7011	0.7147	0.7958	0.7326
	Std	$1.2384 \times 10^{-6}$	0.0267	0.0983	0.0161

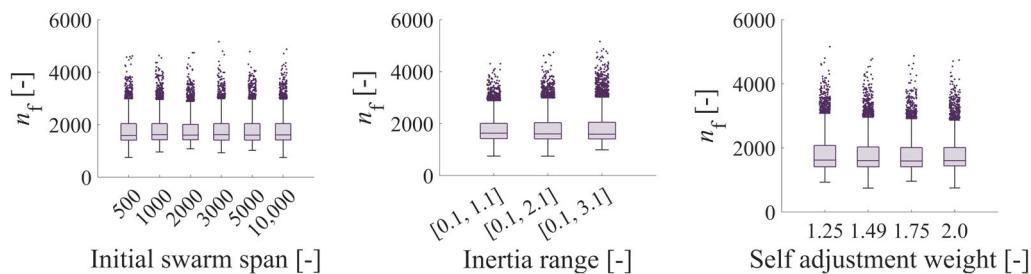
It is clear that the fastest algorithm, based on the average number of expressions of the objective function, is PSCH. This is due to the low number of operations involved in the optimisation [24,46]. In contrast, SURG is slower under the given conditions, as the surrogate function is more complex than the objective function. The GA algorithm, on the other hand, proves to be computationally demanding due to the complexity of the evolutionary algorithm [50]. Additionally, it is noticeable that a small number of SURG settings achieved convergence with fewer evaluations than the minimum shown in Table 7.

As for the ability to find the global extreme, Table 9 shows that the best algorithm is PSWM. Its undeniable advantage over the others is the ability to find values close to the true minimum regardless of the settings. The minimum values of PSWM, PSCH and GA are slightly lower than the global minimum due to the non-compliance with the load capacity limit and flow rate limit. However, the deviation was less than 1%. This is due to Equations (6) and (7), more precisely the exponent, which gives room for a small deviation. On the other hand, the instability of the GA can be noted, as a consequence of which it is necessary to emphasise its correct setting. More precisely, it is the setting of the Gaussian

mutation. Since this single setting results in significant distortion of the results, the values without it are presented in Table 9, and this setting is not considered further.

#### 4.1.1. Particle Swarm

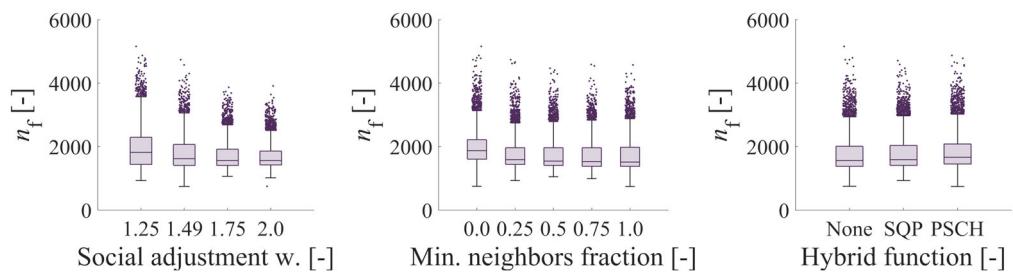
The variance of the objective function values depending on the individual settings is marginal compared to the other algorithms, as shown in Table 9. For this reason, it is appropriate to focus on evaluating the individual settings solely with respect to their computational complexity. The parameter Initial swarm span has a non-negligible influence, even when optimising with boundary conditions. As shown in Figure 3, it is advisable, in terms of computational time, to choose a value of this parameter of either 500 or 10,000. However, it is worth noting that the median for each value of this parameter is roughly the same.



**Figure 3.** Effect of Initial swarm span, minimum neighbours fraction and Inertia range on the number of expressions of the objective function in the case of PSWM using the analytical solution of the journal bearing.

For the Inertia range parameter, we observe a correlation between an increasing range of this parameter and increased computational complexity. Therefore, the most suitable range among the tested values is 0.1–1.1. For the Self adjustment and Social adjustment weight parameters, the recommended value of 1.49 [25,26] appears to be the most effective. For the Self adjustment weight, Figure 3 also shows that doubling the effect of the best position of a given particle has the same effect as the recommended value. For the Social adjustment weight, an increase in the minimum number of evaluations of the objective function is observed as a function of the parameter value. However, except for the value of 1.25, a lower variability is also seen.

A similar situation occurs when selecting the value of the minimum neighbours fraction parameter (see Figure 4). The minimum computation time is achieved when the particles do not interact with each other during the update of the swarm motion (parameter equal to 0) or in the opposite case (parameter equal to 1). However, it should be noted that the first case also exhibits the largest variation in results. Therefore, it cannot be guaranteed that the second variant will not perform significantly better if a more complex bearing model is used.

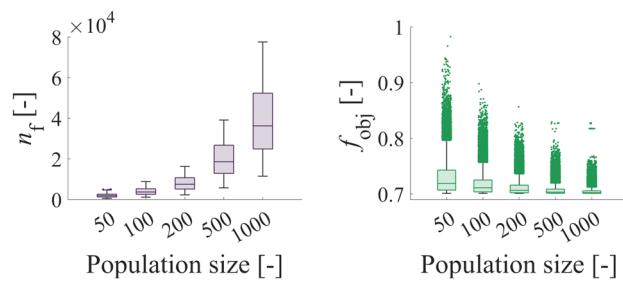


**Figure 4.** Effect of Self adjustment weight, Social adjustment weight and Hybrid function on the number of evaluations of the objective function in the case of PSWM using the analytical solution of the journal bearing.

Similar to the previous algorithm, the Hybrid function was also tested. In the case of using the SQP algorithm, the same low number of evaluations of the objective function could not be achieved as with the other two variants. Figure 4 further shows that the absence of the Hybrid function was less computationally intensive in 50% of the optimisation results with this setup compared to PSCH, which, however, achieved similar minima.

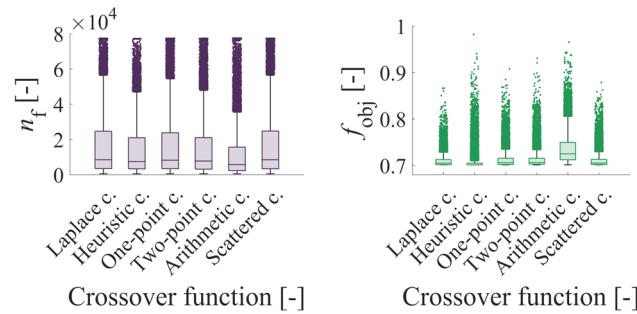
#### 4.1.2. Genetic Algorithm

One of the most important factors influencing the performance of a GA, including both computational complexity and accuracy, is the population size. It is evident from the graphs that, for models with lower computational complexity, using the largest possible population is advisable. However, for more complex models, the computational complexity or the time required for successful optimisation must be considered. As shown in Figure 5, more detailed analysis of the results is necessary due to the variety of different combinations of settings, as the results of some simulations are already out of range. In this case, the results were significantly influenced by other settings.



**Figure 5.** Effect of population size on the number of objective function expressions and its value in the GA case using the analytical solution of the journal bearing.

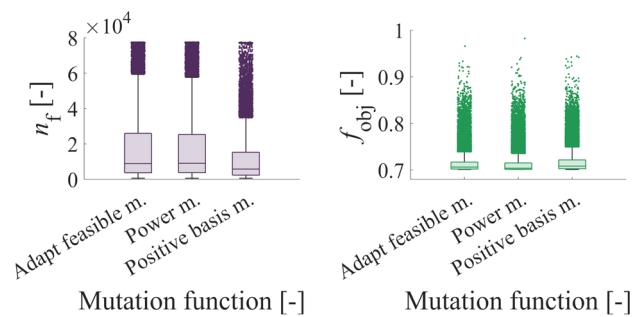
The type of crossover function also had a relatively large influence (see Figure 6). In this case, the results aligned with the literature [58], where the recommended heuristic function significantly improved the algorithm's accuracy compared to others. It is worth noting that it did not significantly increase computational complexity. In contrast, the Arithmetic function, although exhibiting lower complexity, was unable to provide a narrow range of resulting values due to its inherent principles and the way the Crossover function was represented in the next generation design (see Table 5). More specifically, it was the least effective crossover function in terms of achieving optimal objective function values.



**Figure 6.** Effect of the type of crossover function on the number of objective function expressions and its value in the case of GA using the analytical solution of the journal bearing.

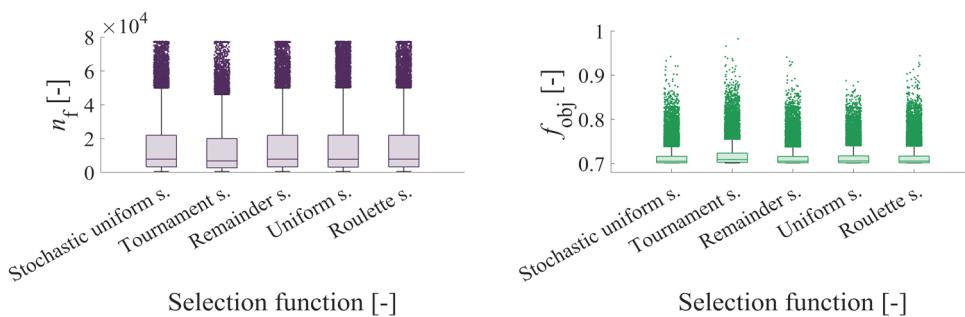
Compared to the crossover function, the mutation function has less influence. This is primarily due to its lower representation in the creation of the next generation. However, Figure 7 shows that the Positive basis mutation results in lower computational complexity at the expense of higher variability in the objective function. On the other hand, Adapt

Feasible mutation and Power mutation achieve similar values of the objective function. Therefore, their effectiveness, in combination with other settings, will be considered in the selection process, similar to [37].



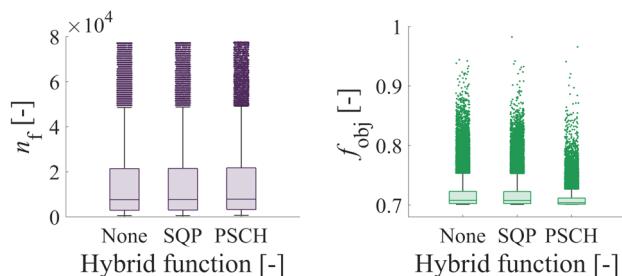
**Figure 7.** Effect of Mutation function type on the number of expressions of the objective function and its value in the case of GA using the analytical solution of journal bearing.

In the case of the selection function (see Figure 8), the commonly recommended tournament selection appears to be suitable primarily when reducing computation time is necessary. Other selection functions achieve comparable computational complexity. Regarding accuracy, due to the low values of the objective function, it was not possible to determine definitively which function is the most suitable. The influence of the selection function type on the ability to find the global extreme and on speed is relatively small.



**Figure 8.** Effect of the selection function on the number of expressions of the objective function and its value in the case of GA using the analytical solution of the journal bearing.

Another important finding is that the application of the Hybrid function has only a marginal effect on computational demand (see Figure 9). However, using PSCH as a secondary algorithm to refine the results, specifically to find the local minimum, appears to be an obvious choice.

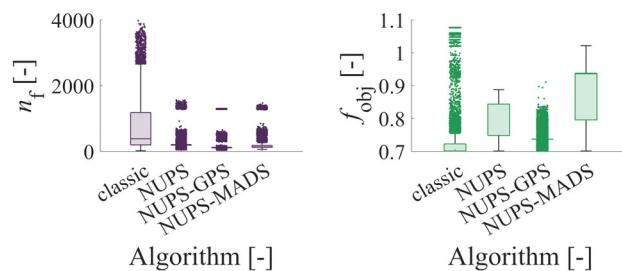


**Figure 9.** Effect of Hybrid function type on the number of expressions of the objective function and its value in the GA case using the analytical solution of journal bearing.

The last parameter tested in the GA was fitness scaling. This is the only parameter tested in this algorithm where the differences between the functions are marginal. For this reason, it was not possible to unambiguously determine the best-fitting option.

#### 4.1.3. Pattern Search

The most important setting for this algorithm is the algorithm selection. This is due to the specificity of the algorithm and the associated limitations of the software used for the optimisations. If one of the variations of the NUPS algorithm is selected, the Polling method, poll ordering algorithm, complete poll and rotate mesh cannot be specified. These algorithms run in a cycle of 16 iterations. In the case of NUPS, it combines two Polling methods, GPS and MADS, whereas NUPS-GPS uses only GPS, and NUPS-MADS uses only OrthoMADS. The data presented in Figure 10 indicate that these algorithms are significantly less computationally intensive. The distances between some quartiles were so small that some outliers were removed from the graph for clarity. For instance, the difference between Q2 and Q3 in the case of NUPS-MADS is approximately 0.2%. Given this and the substantial variation in  $f_{\text{obj}}$ , one might expect significantly worse results compared to the others. Similarly, but with less variation, is NU/S, where  $Q_2 \approx Q_3$ . Relatively unexpected results were obtained for NUPS-GPS, where convergence occurred in 99% of the cases for the simple bearing model. Considering the low variability in  $f_{\text{obj}}$  and the fact that  $Q_1 \approx Q_2$ , the classic algorithm appears to be the best choice. However, due to the low computational complexity, a scenario where NUPS-GPS is used to locate the region of the global minimum, followed by another optimisation algorithm for refinement, can be recommended.



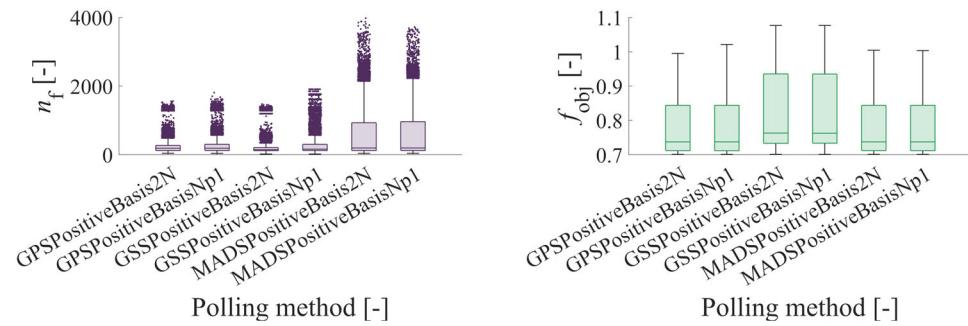
**Figure 10.** Effect of algorithm type on the number of expressions of the objective function and its value in the case of PSCH using the analytical solution of journal bearing.

In contrast, the Cache and rotate mesh parameters have a negligible effect on calculation accuracy. The difference between the quartiles of the objective function values for the given setting variations was below 1%. However, an observable difference emerged in the number of evaluations of the objective function, with a difference below 2.6% for Q3 and slightly less for Q2, as shown in Table 10. Assuming similar results with a more complex bearing model, it can be concluded that rotate mesh has a negligible impact on computational complexity, whereas enabling the Cache to store already evaluated positions may reduce computational complexity.

**Table 10.** Dependence of the objective function value and the number of its evaluations on the Cache and rotate mesh settings.

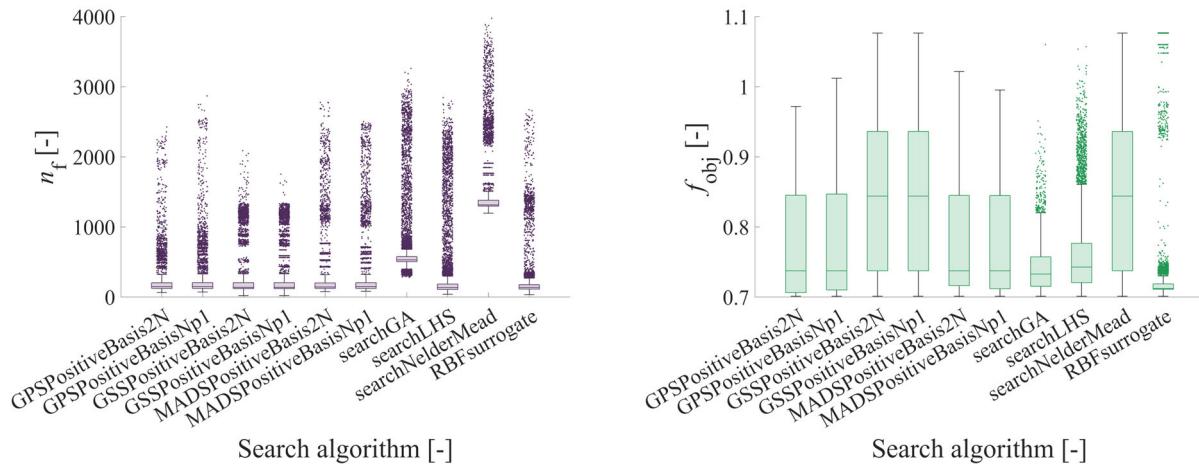
Setting	Cache		Rotate Mesh	
	On	Off	On	Off
Median objective function value	0.7374	0.7374	0.7374	0.7374
Maximum of function evaluations—Q1	113	123	123	123
Maximum of function evaluations—Q2	172	201	189	179

Another important setting in the case of the classic algorithm is the Polling method. The MADS variants are the most computationally intensive (see Figure 11). In contrast, GSSPositiveBasisNp1, which results in the lowest variability in the number of objective function evaluations, exhibits the highest variability in the objective function's value. Given these findings, the GPS Polling method appears to be the most appropriate, as the difference between the two variants in terms of the parameters of interest is negligible.



**Figure 11.** The effect of the Polling method on the number of objective function evaluations and its value in the case of PSCH using the analytical solution of the journal bearing.

The final setting that fundamentally affects how the PSCH operates is the search algorithm (see Figure 12). Although all the tested search algorithms were able to achieve similar minima for the objective function during optimisation, most of them exhibited large variance between quartiles. The search algorithm using the radial basis function (RBFsurrogate), rather than the direct search algorithm, achieved the lowest variability. Considering computational complexity, this search algorithm also appears to be the most appropriate. However, all tested search algorithms, except for searchGA and searchNelderMead, were able to achieve low Q2 values and small variance.



**Figure 12.** The effect of the type of search algorithm on the number of objective function evaluations and its value in the case of PSCH using the analytical solution for the journal bearing.

The poll ordering algorithm, complete poll and complete search do not have a significant effect on the obtained values of the objective function. However, it cannot be guaranteed that this will hold for a more complex bearing model, so the quartiles of the maximum number of objective function evaluations (Table 11) were considered when choosing the settings.

**Table 11.** Effect of the poll ordering algorithm, complete poll and complete search on the number of objective function evaluations in the PSCH case using the analytical solution of the journal bearing.

Setting	Poll Ordering Algorithm			Complete Poll		Complete Search	
	Random	Success	Consecutive	True	False	True	False
Median objective function value [—]	0.7374	0.7374	0.7374	0.7374	0.7374	0.7374	0.7374
Maximum of function evaluations—Q1	123	123	123	123	123	123	123
Maximum of function evaluations—Q2	186	184	179	190	177	189	176
Maximum of function evaluations—Q3	501	501	501	503	500	505	499

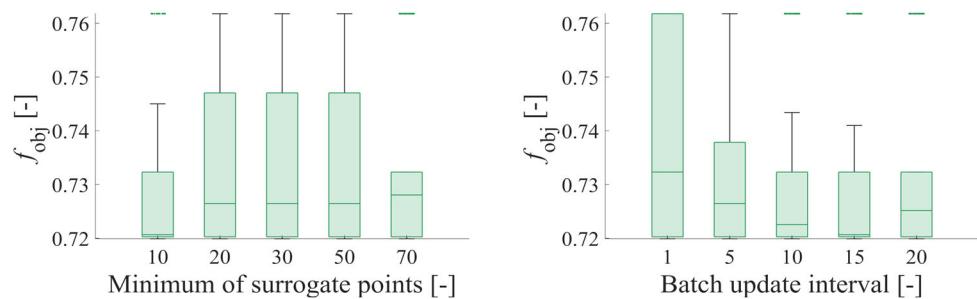
#### 4.1.4. Surrogate

Unlike the previous algorithms, only a small number of settings were tested, and the maximum of function evaluations parameter directly determines the number of evaluations of both the objective and surrogate functions. For this setting, a decrease in the resulting value of the objective function can be observed, except for the highest value (see Table 12). A similar trend can be expected for a more complex bearing model; therefore, the value of this parameter cannot be determined unambiguously.

**Table 12.** Effect of the maximum of function evaluations parameter on the number of objective function expressions in the PSCH case using the analytical solution of the journal bearing.

Maximum of function evaluations	200	500	1000	1500
Median objective function value	0.7618	0.7323	0.7199	0.7206

In contrast, the minimum of surrogate points and Batch update interval settings did not show significant changes in computational complexity with varying tested values (see Figure 13). However, the former achieved significantly less variability in the resulting objective function values when at least 70 points were used to construct the surrogate function. The median was close to the first quartile when 10 points were used, suggesting that this might be the best choice for a more complex bearing model. Similarly, for the Batch update interval setting, the optimal choice was when the replacement function was updated after 15 points.



**Figure 13.** Effect of the minimum of surrogate points and Batch update interval parameters on the number of objective function expressions in the SURG case using the analytical solution of the journal bearing.

#### 4.2. Analysis of the Properties of Optimisation Algorithms Based on the Choice of Initial Distribution of Individuals

Based on the analysis of the optimisation algorithm settings using the analytical solution of HD lubrication (see Appendix A.1) and the values of the output parameters,

the best settings for each algorithm with respect to the given problem were identified. The optimal settings for PSWM are presented in Table 13, the GA settings in Table 14, the PSCH settings in Table 15 and the SURG settings in Table 16. This selection emphasises the ability to find the global minimum value of the objective function with the fewest evaluations of the objective function.

**Table 13.** The result efficient settings of PSWM.

Option	Value
Initial swarm span	500
Minimum neighbours fraction	0
Inertia range	0.1–1.1
Self adjustment weight	1.25
Social adjustment weight	1.49
Swarm size	100
Hybrid function	SQP
Function tolerance	$10^{-6}$
Max. Iterations	600
Max. Stall Iterations	20

**Table 14.** The result efficient settings of GA.

Option	Value
Population size	1000
Fitness scaling	Rank fitness scaling
Elite fraction	0.05
Crossover fraction	0.7
Crossover function	Heuristic crossover
Mutation function	Mutationadaptfeasible
Selection function	Remainder selection
Hybrid function	SQP
Max. Generations	80
Max. Stall Generations	10
Function tolerance	$10^{-6}$
Max. Stall Time	300

**Table 15.** The result efficient settings of PSCH, where  $n_{var}$  is the number of design parameters used for optimisation.

Option	Value
Algorithm	classic
Cache	on
Mesh rotate	on
Poll method	GPSPositiveBasis2N
Search function	MADSPositiveBasis2N
Use complete poll	false
Use complete search	true
Poll order algorithm	consecutive
Max. Iterations	$100n_{var}$
Max. Function Evaluations	$2000n_{var}$
Function tolerance	$10^{-6}$
Step tolerance	$10^{-6}$
Mesh tolerance	$10^{-6}$

**Table 16.** The result efficient settings of SURG.

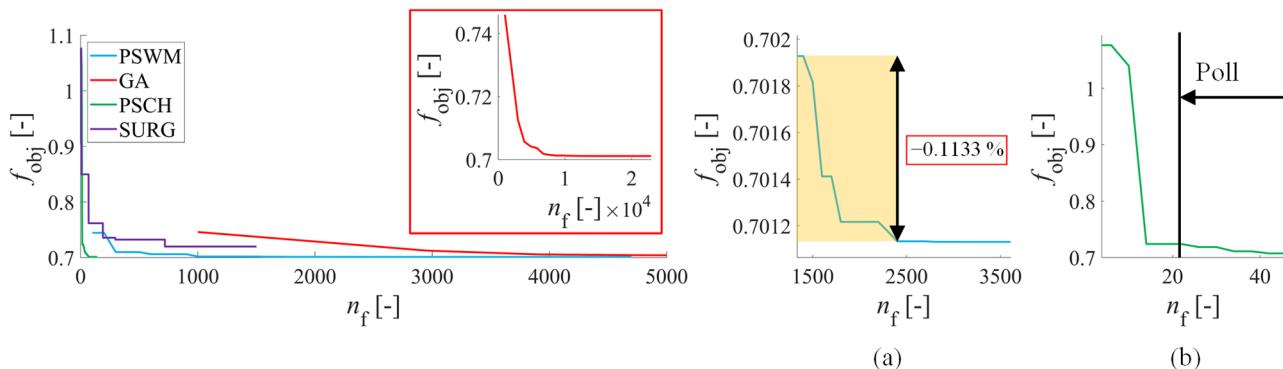
Option	Value
Minimum surrogate points	70
Batch update interval	15
Maximum of function evaluations	1500

The performance of the best algorithms was then compared again on the journal bearing optimisation problem using an analytical solution. To obtain the most accurate results, 100 optimisations were performed for each setting. This approach reduced the influence of the random seed and verified whether it affected a given algorithm with a specific setting. The results comparing the individual best algorithm settings are summarised in Table 17.

**Table 17.** Results for all algorithms with optimal settings.

Function	-	PSWM	GA	PSCH	SURG
Number of objective function evaluations	Min	3977	20,141	142	1500
	Max	7477	26,791	142	1500
	Mean	5580	23,311	142	1500
	Std	724.1141	1185.8935	0.0000	0.0000
Objective function value	Min	0.7011	0.7011	0.7011	0.7199
	Max	0.7011	0.7011	0.7011	0.7199
	Mean	0.7011	0.7011	0.7011	0.7199
	Std	$4.5851 \times 10^{-8}$	$1.1549 \times 10^{-8}$	$1.1159 \times 10^{-16}$	$1.1159 \times 10^{-16}$

In the case of PSWM, the importance of the algorithm's stopping conditions is evident, as starting from  $n_f = 2400$ , there was only a slight change in the value of the objective function (see Figure 14a). Therefore, the possibility of reducing the value of Max. Stall Iterations is suggested. However, it cannot be guaranteed that the optimisation will not stop prematurely. It would therefore be advisable to include an option to save the swarm distribution at the last iteration, which could then be used as input for further optimisation, either when refinement of the results is required or when boundary conditions are marginally changed. The speed of PSWM was strongly dependent on the swarm size parameter, as it determines the minimum number of objective function evaluations.

**Figure 14.** Comparison of progression of objective function values for each optimisation algorithm, with detail (a) showing the region with minimal changes in  $f_{obj}$  (PSWM) and detail (b) showing the stagnation region of the search function (PSCH).

The GA reliably found the global minimum but was the slowest computationally in this case, primarily because the objective function must be evaluated for all individuals in

the population. With a population of 1000 individuals, the GA found the global minimum after an average of 18 generations.

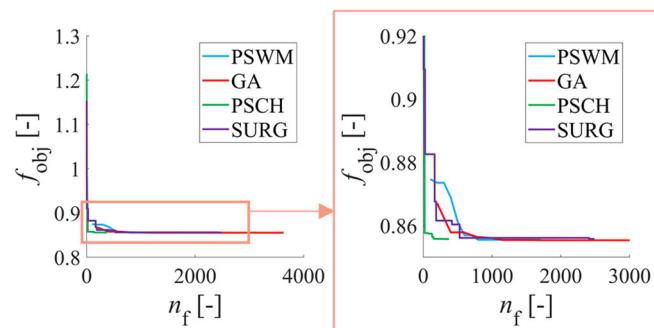
SURG was unable to find the global minimum within the predefined maximum number of iterations listed in Table 16. Since the last improvement occurred approximately halfway through the optimisation, significant improvement cannot be expected without a dramatic increase in the maximum number of function evaluations. However, the minimum found was not far from the global minimum, and in practical cases, the difference is negligible.

The results show that PSCH finds the global minimum with the least computational effort and the fastest. From the progress of the optimisations for the given algorithms with the specified settings in Figure 14, it can be concluded that PSCH quickly identifies the region of the local minimum, typically after only a few evaluations of the objective function (usually after  $n_f = 14$ ), and then spends most of the time searching for the minimum itself. The transition from search to poll is clearly visible in Figure 14b.

#### 4.3. Application of Optimisation Algorithms to Numerically Solved Problems

The optimisation algorithms, using the best settings identified from the analysis based on the analytical solution of HD lubrication (see Appendix A.1), were subsequently applied to optimise HD bearings with the objective function defined by the numerical solution (see Appendices A.2 and A.3). In practical problems, the numerical solution is commonly used, and its primary characteristic is a significantly higher computational complexity in evaluating the objective function. Furthermore, as seen from Equation (A9), cavitation introduces a degree of nonlinearity into the problem, thereby affecting the optimisation space.

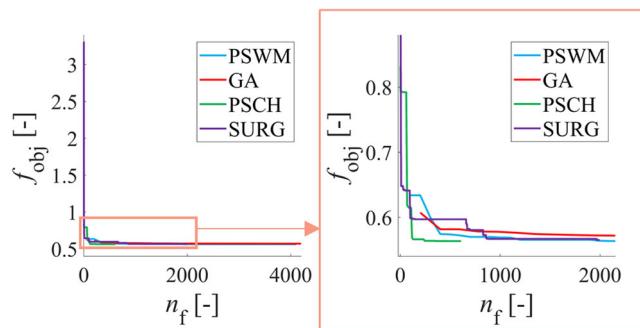
In the case of optimising a journal bearing with three input parameters (see Figure 15), the properties discussed in the previous sections were generally maintained. All the analysed algorithms reliably identified the same minimum, with negligible differences in the objective function. It is important to note that this minimum is not necessarily a global minimum. Due to the computational complexity, it was not found manually. However, significant differences were observed when comparing the speed of the algorithms. PSCH identified the minimum the fastest (finished after  $n_f = 375$ , but managed to identify the region of the local minimum after  $n_f = 153$ ), while the PSWM (finished after  $n_f = 1700$ , but managed to identify the region of the local minimum after  $n_f = 800$ ), GA (finished after  $n_f = 3630$ , but managed to identify the region of the local minimum after  $n_f = 1730$ ) and SURG (finished after  $n_f = 2490$ ) algorithms showed a balanced performance in terms of speed, as measured by the number of objective function evaluations.



**Figure 15.** Comparison of the objective function value progress for each optimisation algorithm in the optimisation of a journal bearing with three variables.

A numerical computational model of the HD thrust bearing with five input parameters was also used to compare the algorithms. This model incorporates additional nonlinear-

ties, such as variable lubricant properties (viscosity, density and specific heat capacity), two-phase fluid dynamics, cavitation and turbulence, resulting in a more complex optimisation space. The optimisation progress (see Figure 16) again highlights the effectiveness of PSCH (finished after  $n_f = 608$ , but managed to identify the region of the local minimum after  $n_f = 481$ ), which found the minimum and was also the fastest. The PSWM (finished after  $n_f = 4100$ , but managed to identify the region of the local minimum after  $n_f = 2500$ ) and SURG (finished after  $n_f = 1995$ ) algorithms were essentially equal in both finding the minimum (difference under 0.32%) and in speed. However, the GA failed to find the corresponding minimum ( $f_{\text{obj}} = 0.5707$ , in comparison to PSWM  $f_{\text{obj}} = 0.5634$ ), even after many thousands of evaluations of the objective function (finished after  $n_f = 4200$ ), proving inefficient in this case.



**Figure 16.** Comparison of the progression of objective function values for individual optimisation algorithms in the case of optimising a thrust bearing with five variables.

## 5. Conclusions

The PSWM, GA, PSCH and SURG algorithms for optimising HD bearing parameters were analysed across three types of problems, involving both analytical and numerical calculations of the objective function. The results indicated that the PSCH algorithm was the most efficient in all cases, both in terms of finding the global minimum and in speed. The PSCH algorithm achieved the following results in the tasks tested:

- Analytical solution of the journal HD bearing:  $n_f = 142$  and  $f_{\text{obj}} = 0.7011$ ;
- Numerical solution of the journal HD bearing:  $n_f = 375$  and  $f_{\text{obj}} = 0.8557$ ;
- Numerical solution of the thrust HD bearing:  $n_f = 608$  and  $f_{\text{obj}} = 0.5634$ .

The PSWM also reliably found the global minimum but was slower on the defined problems. In this case, the results were as follows:

- Analytical solution of the journal HD bearing:  $n_f = 4700$  and  $f_{\text{obj}} = 0.7011$ ;
- Numerical solution of the journal HD bearing:  $n_f = 1700$  and  $f_{\text{obj}} = 0.8554$ ;
- Numerical solution of the thrust HD bearing:  $n_f = 4100$  and  $f_{\text{obj}} = 0.5632$ .

The GA and SURG algorithms were less efficient in the tested problems; in some cases, they did not find the global minimum with sufficient accuracy and exhibited slower search speeds. This is mainly due to the fact that the number of objective function evaluations of the GA algorithm is highly dependent on the size of the population. Meanwhile, the performance of the SURG algorithm is highly dependent on the complexity of the original function and the surrogate function. To be more specific, the results in the case of the GA algorithm were as follows:

- Analytical solution of the journal HD bearing:  $n_f = 22,900$  and  $f_{\text{obj}} = 0.7011$ ;
- Numerical solution of the journal HD bearing:  $n_f = 3630$  and  $f_{\text{obj}} = 0.8553$ ;
- Numerical solution of the thrust HD bearing:  $n_f = 4100$  and  $f_{\text{obj}} = 0.5707$ ;

And in the case of SURG algorithm, the results were as follows:

- Analytical solution of the journal HD bearing:  $n_f = 1500$  and  $f_{obj} = 0.7199$ ;
- Numerical solution of the journal HD bearing:  $n_f = 2490$  and  $f_{obj} = 0.8558$ ;
- Numerical solution of the thrust HD bearing:  $n_f = 1995$  and  $f_{obj} = 0.5650$ .

Based on the obtained results, it can be concluded that PSWM is the best default choice, despite the fact that the PSCH algorithm was the most efficient. PSCH can also be recommended for scenarios with lower computational requirements, but it is necessary to verify which search algorithm and Polling method are most suitable. The efficiency of PSCH was significantly affected by the choice of search algorithm and Polling method, to the extent that some combinations led to convergence at a value that was not the global minimum. This occurred when using GSS as the search algorithm.

The PSWM demonstrated very good efficiency in the proposed problems, consistently finding the global minimum, although in some cases at a slower rate. A key advantage of this algorithm is the minimal effect of the tested settings on its ability to find the global minimum. The drawback, however, is the slower rate of convergence, primarily due to the higher number of objective function evaluations, which is influenced by the swarm size parameter. Consequently, the computational complexity of PSWM is directly dependent on the swarm size.

Although widely used in the literature for optimising HD bearings, the GA showed a reduced ability to find the global minimum in the tested problems and also demonstrated very slow global minimum-finding speeds. Therefore, the computational complexity of the GA is directly dependent on the population size. It can be recommended for optimisation only when the previous two algorithms are unsuitable.

SURG is directly dependent on the specified number of evaluations of the objective function. Therefore, its use is less suitable for HD bearing optimisation with the presented computational models. However, if the user defines a significantly higher maximum number of evaluations (for example,  $2 \times 10^4$  for the HD models presented here) and manually stops the optimisation based on the objective function's progress, finding the global minimum is not guaranteed.

The analysed algorithms were tested to find the optimal parameters for HD bearings, assuming that the optimisation space exhibits similar characteristics. If the optimisation space differs significantly, such as when optimising other types of problems, the efficiency of the individual algorithms presented here, along with the best settings found, is not guaranteed.

The presented findings can be applied to the optimisation of HD bearings as well as to tasks with similar complexity or objective functions. Additionally, the knowledge gained can serve as a foundation for optimising more complex bearing models. By using the presented approach and insights, the desired bearing properties can be achieved in a relatively short time.

**Author Contributions:** Conceptualization, P.N.; methodology, P.N.; software, F.K. and P.N.; validation, P.N.; formal analysis, F.K.; investigation, F.K.; resources, F.K. and P.N.; data curation, F.K.; writing—original draft preparation, F.K.; writing—review and editing, P.N.; visualization, F.K.; supervision, P.N.; project administration, F.K. and P.N.; funding acquisition, P.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This publication was supported by the project “Innovative Technologies for Smart Low Emission Mobilities”, funded as project No. CZ.02.01.01/00/23\_020/0008528 by Programme Johannes Amos Comenius, call Intersectoral cooperation.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Glossary

### Nomenclature

$b$	bearing width
$d$	bearing diameter
$e$	shaft eccentricity
$h$	lubrication layer thickness
$\dot{m}$	lubrication mass flow rate
$n$	rotor speed
$p$	hydrodynamic pressure
$r$	radius
$t$	time
$\mathbf{x}$	vector of design parameters
$x, y, z$	coordinates
$A$	area
$C$	oil heat capacity
$U$	linear speed on the rotor surface
$a_{j,1}; a_{j,2}$	first and second parent
$c_{\text{ax}}$	axial clearance
$c_d$	diametral bearing clearance
$c_{j,1}; c_{j,2}$	first and second children
$f_{\text{obj}}$	objective function
$h_{0ts}$	minimum thickness of the lubrication gap on the thrust side of the bearing
$h_g$	groove height
$\dot{m}_{\text{lim}}$	mass flow rate limit of journal (thrust) bearing
$\dot{m}_o$	lubrication flow rate at outlet
$n_f$	number of objective function evaluations
$n_p$	number of pads
$n_{\text{var}}$	number of optimised variables
$p_{\text{cav}}$	saturation pressure of the oil vapor
$p_h$	hydrodynamic pressure
$p_{\text{in}}$	inlet lubricant pressure
$p_{\text{out}}$	outlet lubricant pressure
$r_0$	inner radius
$r_1$	outer radius
$r_{\text{tmax}}$	taper outer radius
$r_{\text{tmin}}$	taper inner radius
$s_{\dot{m}}$	flow rate correction factor
$s_F$	load capacity correction factor
$s_M$	friction torque ratio
$u_1$	velocity of the pin
$u_2$	velocity of the pan
$v_R$	radial pin velocity
$A_{\text{in}}$	lubricant inlet area
$A_{\text{out}}$	lubricant outlet area
$A_{\text{sid}}$	area of lubricant side outlet
$F_c$	load carrying capacity
$F_{\text{lim}}$	load carrying capacity limit of journal (thrust) bearing
$G_\alpha, G_\beta$	turbulent flow correction coefficients
$M_{f,\text{ref}}$	friction torque of the reference bearing
$M_f$	friction torque

$T_{in}$	oil temperature at inlet
$\varphi_t$	taper angle
$\varepsilon$	relative eccentricity
$\mu$	dynamic lubricant viscosity
$\rho$	oil density
$\varphi$	angle
$\omega$	angular velocity of the shaft (thrust ring)
$\beta_j$	random number generated from the Laplace distribution
$\gamma_t$	wedge taper angle
$\mu_{mix}$	dynamic viscosity of oil-air mixture
$\rho_{mix}$	density of oil-air mixture
$\varphi_g$	groove angular position
<b>Abbreviations</b>	
2D	two-dimensional
3D	three-dimensional
CFD	computational fluid dynamics
GA	genetic algorithm
GPS	generalised pattern search
GSS	generating set search
HD	hydrodynamic
LCC	load carrying capacity
MADS	mesh adaptive direct search
nups	nonuniform pattern search
OrthoMADS	orthogonal mesh adaptive direct search
PSCH	pattern search
PSWM	particle swarm
Q1, Q2, Q3	first quartile, median (second quartile) and third quartile
SQP	sequential quadratic programming
Std	standard deviation
SURG	surrogate

## Appendix A. Computational Models of Bearing Lubrication

The study includes two types of computational models for bearing lubrication solution. The analytical model according to short bearing theory is used in Sections 4.1 and 4.2. Numerical computational models are used in the case of Section 4.3. All the computational models used in this study use the same definition of the objective function according to Equation (4).

### Appendix A.1. Computational Model of Journal Bearing According to Short Bearing Theory

Bearing lubrication involves the flow of a viscous fluid in the lubrication gap. In general, this is a 3D flow, and it is necessary to solve the transport equations that describe the conservation of mass, momentum and energy. By introducing simplifying assumptions, the basic Reynolds equation for HD lubrication in thin layers can be derived. With additional assumptions, this equation can be further simplified. For journal bearings, where the axial length is smaller than the shaft diameter, the pressure gradient along the axis of rotation is much larger than the pressure gradient in the circumferential direction. Thus, the short bearing theory can be applied [1]. This approximation gives reasonably good results for  $\frac{b}{d} < \frac{1}{3}$ .

The vector of design parameters for optimisation ( $\mathbf{x}$ ) that forms the optimisation space is defined by Equation (2).

For comparison of optimisation algorithms, the bearing LCC in accordance with short bearing theory [1] is determined as follows:

$$F_c = \frac{\mu\omega db^3}{c_d^2} \frac{\varepsilon\pi}{2(1-\varepsilon^2)^2} \sqrt{\left(\frac{16}{\pi^2} - 1\right)\varepsilon^2 + 1}, \quad (\text{A1})$$

where  $\omega$  is the angular velocity of the pin rotation relative to the shell. The friction moment is calculated from the relation [1]

$$M_f = \frac{\pi\mu\omega bd^3}{2c_d} \frac{1}{(1-\varepsilon^2)^{0.5}}, \quad (\text{A2})$$

and the lubrication mass flow rate is defined as [1]

$$\dot{m} = \frac{1}{4}\rho\omega dc_d b\varepsilon. \quad (\text{A3})$$

For the optimisation of the journal bearing variables, only operating conditions No. 3, according to Table 3, will be used.

#### Appendix A.2. Numerical Computational Model of Journal Bearing

In the case of a more detailed solution of HD lubrication for a journal bearing, the Reynolds equation, assuming 2D flow of incompressible fluid, is solved numerically. However, for the purposes of this paper, the simplified nonlinear numerical solution presented by Novotný et al. [7] was used. The algorithm has been verified in many previous projects using 3D CFD methods and experimental measurements. There is of course a certain inaccuracy, but this inaccuracy has absolutely no effect on the study presented here. The basic Reynolds equation describing the pressure distribution in the lubricating layer is [7]

$$\frac{\partial}{\partial x} \left( h^3 \frac{\partial p_h}{\partial x} \right) + \frac{\partial}{\partial z} \left( h^3 \frac{\partial p_h}{\partial z} \right) = 6\mu \left[ (u_1 + u_2) \frac{\partial h}{\partial x} + 2 \frac{\partial h}{\partial t} \right], \quad (\text{A4})$$

where  $x$  and  $z$  are the coordinates,  $t$  is time,  $h$  is the thickness of the local lubrication layer,  $p_h$  is the HD pressure, and  $u_1$  and  $u_2$  are the velocities of the pin and the pan, respectively. Equation (A4) is supplied with boundary conditions defining pressure conditions at the inlet and outlet areas, as follows [7]:

$$p_h = \begin{cases} 0 & \text{for } A \in \{A_{out}, A_{sid}\} \\ p_{in} & \text{for } A \in \{A_{in}\} \end{cases}, \quad (\text{A5})$$

where  $p_{in}$  is the inlet lubricant pressure,  $A_{in}$  is the lubricant inlet area,  $A_{out}$  is the lubricant outlet area,  $A_{sid}$  is the area of the lubricant side outlet, and  $A$  is the area. The given solution is complemented by a simplified cavitation condition defined as  $p_h \geq 0$ . The Reynolds Equation (A4) is transformed into dimensionless quantities and is numerically solved as a steady-state problem. The numerical computational model of journal bearing adopted from Novotný et al. [7] uses a finite difference method, and in the case of this study, a  $160 \times 120$ -point grid was used. A detailed description of the solution of the Reynolds equation is provided in Novotný et al. [7].

Moreover, the integral values of the bearing, including bearing LCC ( $F_c$ ), friction moment ( $M_f$ ) and lubricant flow ( $\dot{m}_o$ ), are obtained numerically based on the known pressure distribution in the lubrication layer and the geometry of the lubrication layer [7].

The vector of selected design parameters for optimisation ( $\mathbf{x}$ ) that forms the optimisation space is identical to the space considered in the analytical solution of the journal

bearing and is given by Equation (2). For the optimisation of the journal bearing performance using the numerical computational model, the set of operating conditions No. 1, 2 and 4, as defined in Table 3, was used.

#### Appendix A.3. Numerical Computational Model of Thrust Bearing

The thrust bearing computational model used in this paper was presented by Novotný and Hrabovský [59] and is significantly more detailed, as it includes the variable properties of the lubricant, the oil–air mixture, the effect of centrifugal forces on the lubrication layer and the effect of energy transfer to the surrounding walls. This experimentally verified 2D model extends the well-known thin lubrication layer theory by correcting for the effects of lubricant temperature, inertial forces and turbulence. The lubricant is assumed to be a mixture of oil and gas bubbles with continuously distributed properties throughout the lubrication gap volume. The model calculates the HD pressure  $p_h(r, \varphi)$ , lubricant dynamic viscosity  $\mu_{\text{mix}}(r, \varphi)$  and lubricant density  $\rho_{\text{mix}}(r, \varphi)$  in the lubricating layer, based on bearing geometry, operating conditions and lubricant properties. The pressures at the inlet ( $p_{\text{in}}$ ) and outlet ( $p_{\text{out}}$ ), the angular velocity of the thrust ring ( $\omega$ ) and the temperature at the inlet ( $T_{\text{in}}$ ) are the operating conditions that must be included. The model also corrects for turbulent effects using coefficients  $G_\alpha$ ,  $G_\beta$ . A schematic of the working surface of a bearing segment is presented in Figure 2.

According to Novotný and Hrabovský [59], the extended Reynolds equation describing the pressure distribution in the lubrication layer is as follows:

$$\frac{\partial}{\partial r} \left( \frac{G_\alpha \rho_{\text{mix}} r h^3}{\mu_{\text{mix}}} \left( \frac{\partial p_h}{\partial r} + 0.3 \rho_{\text{mix}} r \omega^2 \right) \right) + \frac{1}{r} \frac{\partial}{\partial \varphi} \left( \frac{G_\beta \rho_{\text{mix}} h^3}{\mu_{\text{mix}}} \frac{\partial p_h}{\partial \varphi} \right) - \frac{r \omega}{2} \frac{\partial (\rho_{\text{mix}} h)}{\partial \varphi} - r \frac{\partial (\rho_{\text{mix}} h)}{\partial t} = 0. \quad (\text{A6})$$

Equation (A7) is supplemented by boundary conditions in the form

$$p_h = p_{\text{in}} \text{ for } A \in \{A_{\text{in}}\} \text{ and} \quad (\text{A7})$$

$$p_h = p_{\text{out}} \text{ for } A \in \{A_{\text{out}}\} \quad (\text{A8})$$

and the following defined cavitation condition is also necessary:

$$p_h \geq p_{\text{cav}}. \quad (\text{A9})$$

where  $p_{\text{cav}}$  is the saturation pressure of the oil vapor, with a value of 2320 Pa.

Based on the calculated pressure distribution in the lubrication layer and the geometry of the lubrication layer, the integral characteristics of the thrust HD bearing, including LCC ( $F_c$ ), friction moment ( $M_f$ ) and lubricant flow ( $\dot{m}_o$ ), are calculated. The procedure for calculating these characteristics can be found in Novotný and Hrabovský [59]. The chosen model is a suitable compromise, as it provides a physically deeper description of HD lubrication, while also allowing for a relatively fast solution. The numerical computational model of the thrust bearing adopted from Novotný and Hrabovský [59] once again uses the finite-difference method, but in this case, it is assumed to be a mass-conservative algorithm considering two-phase flow. Again, a grid with a resolution of  $160 \times 120$  points was used for the numerical solution.

The vector of design parameters for the optimisation of the thrust bearing is defined by Equation (3).

For the optimisation of the thrust bearing performance using the numerical computational model, the set of operating conditions No. 1, 2, 3 and 4, as defined in Table 3, was used.

## References

1. Dubois, G.B.; Ocvirk, F.W. Analytical Derivation and Experimental Evaluation of Short-Bearing Approximation for Full Journal Bearing. 1953. Available online: <https://ntrs.nasa.gov/citations/19930092184> (accessed on 22 July 2024).
2. Fouflias, D.G.; Charitopoulos, A.G.; Papadopoulos, C.I.; Kaiktsis, L. Thermohydrodynamic Analysis and Tribological Optimization of a Curved Pocket Thrust Bearing. *Tribol. Int.* **2017**, *110*, 291–306. [CrossRef]
3. Zouzoulas, V.; Papadopoulos, C.I. 3-D Thermohydrodynamic Analysis of Textured, Grooved, Pocketed and Hydrophobic Pivoted-Pad Thrust Bearings. *Tribol. Int.* **2017**, *110*, 426–440. [CrossRef]
4. Charitopoulos, A.; Visser, R.; Eling, R.; Papadopoulos, C. Design Optimization of an Automotive Turbocharger Thrust Bearing Using a Cfd-Based Thd Computational Approach. *Lubricants* **2018**, *6*, 21. [CrossRef]
5. Li, Y.; Huang, W.; Sang, R. Analysis of the Influencing Factors of Aerostatic Bearings on Pneumatic Hammering. *Lubricants* **2024**, *12*, 395. [CrossRef]
6. Novotný, P.; Hrabovský, J.; Juračka, J.; Klíma, J.; Hort, V. Effective Thrust Bearing Model for Simulations of Transient Rotor Dynamics. *Int. J. Mech. Sci.* **2019**, *157–158*, 374–383. [CrossRef]
7. Novotný, P.; Škara, P.; Hliník, J. The Effective Computational Model of the Hydrodynamics Journal Floating Ring Bearing for Simulations of Long Transient Regimes of Turbocharger Rotor Dynamics. *Int. J. Mech. Sci.* **2018**, *148*, 611–619. [CrossRef]
8. Nicoletti, R. Optimization of Journal Bearing Profile for Higher Dynamic Stability Limits. *J. Tribol.* **2013**, *135*, 011702. [CrossRef]
9. Ramos, D.J.; Daniel, G.B. Microgroove Optimization to Improve Hydrodynamic Bearing Performance. *Tribol. Int.* **2022**, *174*, 107667. [CrossRef]
10. Hashimoto, H.; Matsumoto, K. Improvement of Operating Characteristics of High-Speed Hydrodynamic Journal Bearings by Optimum Design: Part I—Formulation of Methodology and Its Application to Elliptical Bearing Design. *J. Tribol.* **2001**, *123*, 305–312. [CrossRef]
11. Zhang, J.; Lu, L.; Zheng, Z.; Tong, H.; Huang, X. Experimental Verification: A Multi-Objective Optimization Method for Inversion Technology of Hydrodynamic Journal Bearings. *Struct. Multidiscip. Optim.* **2023**, *66*, 14. [CrossRef]
12. van Ostayen, R.A.J. Film Height Optimization of Dynamically Loaded Hydrodynamic Slider Bearings. *Tribol. Int.* **2010**, *43*, 1786–1793. [CrossRef]
13. Cheng, C.-H.; Chang, M.-H. The Optimization for The Shape Profile of the Slider Surface Under Ultra-Thin Film Lubrication Conditions by the Rarefied-Flow Model. *J. Mech. Des.* **2009**, *131*, 101010. [CrossRef]
14. Rajan, M.; Rajan, S.D.; Nelson, H.D.; Chen, W.J. Optimal Placement of Critical Speeds in Rotor-Bearing Systems. *J. Vib. Acoust.* **1987**, *109*, 152–157. [CrossRef]
15. Fesanghary, M.; Khonsari, M.M. Topological and Shape Optimization of Thrust Bearings for Enhanced Load-Carrying Capacity. *Tribol. Int.* **2012**, *53*, 12–21. [CrossRef]
16. Saruhan, H.; Rouch, K.E.; Roso, C.A. Design Optimization of Tilting-Pad Journal Bearing Using a Genetic Algorithm. *Int. J. Rotating Mach.* **2004**, *10*, 301–307. [CrossRef]
17. Saruhan, H. Optimum Design of Rotor-Bearing System Stability Performance Comparing an Evolutionary Algorithm Versus A Conventional Method. *Int. J. Mech. Sci.* **2006**, *48*, 1341–1351. [CrossRef]
18. Papadopoulos, C.I.; Nikolakopoulos, P.G.; Kaiktsis, L. Evolutionary Optimization of Micro-Thrust Bearings with Periodic Partial Trapezoidal Surface Texturing. *J. Eng. Gas Turbines Power* **2011**, *133*, 012301. [CrossRef]
19. Novotný, P.; Vacula, J.; Hrabovský, J. Solution Strategy for Increasing the Efficiency of Turbochargers by Reducing Energy Losses in the Lubrication System. *Energy* **2021**, *236*, 121402. [CrossRef]
20. Matsuda, K.; Kanemitsu, Y.; Kijimoto, S. Optimal Clearance Configuration of Fluid-Film Journal Bearings for Stability Improvement. *ASME J. Tribol.* **2004**, *126*, 125–131. [CrossRef]
21. Novotný, P.; Jonák, M.; Vacula, J. Evolutionary Optimisation of the Thrust Bearing Considering Multiple Operating Conditions in Turbomachinery. *Int. J. Mech. Sci.* **2021**, *195*, 106240. [CrossRef]
22. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948. [CrossRef]
23. Mezura-Montes, E.; Coello Coello, C.A. Constraint-Handling In Nature-Inspired Numerical Optimization: Past, Present And Future. *Swarm Evol. Comput.* **2011**, *1*, 173–194. [CrossRef]
24. Pedersen, M.E.H.; Chipperfield, A.J. Simplifying Particle Swarm Optimization. *Appl. Soft Comput.* **2010**, *10*, 618–628. [CrossRef]
25. Eberhart, R.C.; Shi, Y. Particle Swarm Optimization: Developments, Applications and Resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Republic of Korea, 27–30 May 2001; pp. 81–86. [CrossRef]
26. Piotrowski, A.P.; Napiorkowski, J.J.; Piotrowska, A.E. Population Size in Particle Swarm Optimization. *Swarm Evol. Comput.* **2020**, *58*, 100718. [CrossRef]
27. Byrd, R.H.; Gilbert, J.C.; Nocedal, J. A Trust Region Method Based on Interior Point Techniques for Nonlinear Programming. *Math. Program.* **2000**, *89*, 149–185. [CrossRef]

28. Byrd, R.H.; Hribar, M.E.; Nocedal, J. An Interior Point Algorithm for Large-Scale Nonlinear Programming. *SIAM J. Optim.* **1999**, *9*, 877–900. [[CrossRef](#)]
29. Coleman, T.F.; Li, Y. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM J. Optim.* **1996**, *6*, 418–445. [[CrossRef](#)]
30. Waltz, R.A.; Morales, J.L.; Nocedal, J.; Orban, D. An Interior Algorithm for Nonlinear Optimization That Combines Line Search and Trust Region Steps. *Math. Program.* **2006**, *107*, 391–408. [[CrossRef](#)]
31. Torczon, V. On the Convergence of Pattern Search Algorithms. *SIAM J. Optim.* **1997**, *7*, 1–25. [[CrossRef](#)]
32. Ankenbrandt, C.A. An Extension to the Theory of Convergence and a Proof of the Time Complexity of Genetic Algorithms. In *Foundations of Genetic Algorithms*; Elsevier: Amsterdam, The Netherlands, 1991; pp. 53–68. ISBN 9780080506845.
33. Alander, J.T. On Optimal Population Size of Genetic Algorithms. In Proceedings of the CompEuro 1992 Proceedings Computer Systems and Software Engineering, Hague, The Netherlands, 4–8 May 1992; pp. 65–70.
34. Sadjadi, F.; Javidi, B.; Psaltis, D. Comparison of Fitness Scaling Functions in Genetic Algorithms with Applications to Optical Processing. In Proceedings of the Optical Science and Technology, the SPIE 49th Annual Meeting, Denver, CO, USA, 2–6 August 2004; pp. 356–364. [[CrossRef](#)]
35. Mishra, A.; Shukla, A. Analysis of the Effect of Elite Count on the Behavior of Genetic Algorithms: A Perspective. In Proceedings of the 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 5–7 January 2017; pp. 835–840.
36. Yang, H.; Su, M.; Wang, X.; Gu, J.; Cai, X. Particle Sizing with Improved Genetic Algorithm by Ultrasound Attenuation Spectroscopy. *Powder Technol.* **2016**, *304*, 20–26. [[CrossRef](#)]
37. Deep, K.; Singh, K.P.; Kansal, M.L.; Mohan, C. A Real Coded Genetic Algorithm for Solving Integer and Mixed Integer Optimization Problems. *Appl. Math. Comput.* **2009**, *212*, 505–518. [[CrossRef](#)]
38. Vahdati, G.; Yaghoubi, M.; Poostchi, M.; Naghibi-Sistani, M.B. A New Approach to Solve Traveling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator. In Proceedings of the 2009 International Conference of Soft Computing and Pattern Recognition, Malacca, Malaysia, 4–7 December 2009; pp. 112–116.
39. Deep, K.; Thakur, M. A New Mutation Operator for Real Coded Genetic Algorithms. *Appl. Math. Comput.* **2007**, *193*, 211–230. [[CrossRef](#)]
40. Wright, A.H. Genetic Algorithms for Real Parameter Optimization. In *Foundations of Genetic Algorithms*; Elsevier: Amsterdam, The Netherlands, 1991; pp. 205–218. ISBN 9780080506845.
41. Köksoy, O.; Yalcinoz, T. Robust Design Using Pareto Type Optimization: A Genetic Algorithm with Arithmetic Crossover. *Comput. Ind. Eng.* **2008**, *55*, 208–218. [[CrossRef](#)]
42. Hinterding, R. Gaussian Mutation and Self-Adaption for Numeric Genetic Algorithms. In Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, Perth, Australia, 29 November–1 December 1995; p. 384. [[CrossRef](#)]
43. Abramson, M.A.; Audet, C.; Dennis, J.E.; Digabel, S.L. Orthomads: A Deterministic Mads Instance with Orthogonal Directions. *SIAM J. Optim.* **2009**, *20*, 948–966. [[CrossRef](#)]
44. Goldberg, D.E.; Deb, K. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms*; Elsevier: Amsterdam, The Netherlands, 1991; Volume 1, pp. 69–93. ISBN 9780080506845.
45. Audet, C.; Dennis, J.E. Analysis of Generalized Pattern Searches. *SIAM J. Optim.* **2002**, *13*, 889–903. [[CrossRef](#)]
46. Kolda, T.G.; Lewis, R.M.; Torczon, V. Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Rev.* **2003**, *45*, 385–482. [[CrossRef](#)]
47. Lewis, R.M.; Torczon, V.J.; Kolda, T.G. A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints. *Sandia Rep.* **2006**, *1–45*. [[CrossRef](#)]
48. Lewis, R.M.; Shepherd, A.; Torczon, V. Implementing Generating Set Search Methods for Linearly Constrained Minimization. *SIAM J. Sci. Comput.* **2007**, *29*, 2507–2530. [[CrossRef](#)]
49. Audet, C.; Dennis, J.E. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM J. Optim.* **2006**, *17*, 188–217. [[CrossRef](#)]
50. Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley Professional: Boston, MA, USA, 1989; ISBN 2-201-15767-5.
51. Shang, X.; Chao, T.; Ma, P.; Yang, M. An Efficient Local Search-Based Genetic Algorithm for Constructing Optimal Latin Hypercube Design. *Eng. Optim.* **2020**, *52*, 271–287. [[CrossRef](#)]
52. Lagarias, J.C.; Reeds, J.A.; Wright, M.H.; Wright, P.E. Convergence Properties of The Nelder-Mead Simplex Method in Low Dimensions. *SIAM J. Optim.* **1998**, *9*, 112–147. [[CrossRef](#)]
53. Gutmann, H.-M. A Radial Basis Function Method for Global Optimization. *J. Glob. Optim.* **2021**, *19*, 201–227. [[CrossRef](#)]
54. Jakobsson, S.; Patriksson, M.; Rudholm, J.; Wojciechowski, A. A Method for Simulation Based Optimization Using Radial Basis Functions. *Optim. Eng.* **2010**, *11*, 501–532. [[CrossRef](#)]
55. Regis, R.G.; Shoemaker, C.A. A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions. *INFORMS J. Comput.* **2007**, *19*, 497–509. [[CrossRef](#)]

56. McGill, R.; Tukey, J.W.; Larsen, W.A. Variations of Box Plots. *Am. Stat.* **1978**, *32*, 12–16. [[CrossRef](#)]
57. Cox, N.J. Speaking Stata: Creating and Varying Box Plots. *Stata J. Promot. Commun. Stat. Stata* **2009**, *9*, 478–496. [[CrossRef](#)]
58. Michalewicz, Z. *Genetic Algorithms Data Structures = Evolution Programs*, 3rd revised and extended ed.; Springer: Berlin, Germany, 1996; ISBN 35-406-0676-9.
59. Novotný, P.; Hrabovský, J. Efficient Computational Modelling of Low Loaded Bearings of Turbocharger Rotors. *Int. J. Mech. Sci.* **2020**, *174*, 105505. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.