

Article

High-Fidelity Surrogate Based Multi-Objective Optimization Algorithm

Adel Younis ^{1,*}  and Zuomin Dong ² ¹ Mechanical Engineering Department, Australian University, P.O. Box 1411, Safat 13015, Kuwait² Department of Mechanical Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada

* Correspondence: a.younis@ack.edu.kw; Tel.: +965-553-76807

Abstract: The employment of conventional optimization procedures that must be repeatedly invoked during the optimization process in real-world engineering applications is hindered despite significant gains in computing power by computationally expensive models. As a result, surrogate models that require far less time and resources to analyze are used in place of these time-consuming analyses. In multi-objective optimization (MOO) problems involving pricey analysis and simulation techniques such as multi-physics modeling and simulation, finite element analysis (FEA), and computational fluid dynamics (CFD), surrogate models are found to be a promising endeavor, particularly for the optimization of complex engineering design problems involving black box functions. In order to reduce the expense of fitness function evaluations and locate the Pareto frontier for MOO problems, the automated multiobjective surrogate based Pareto finder MOO algorithm (AMSP) is proposed. Utilizing data samples taken from the feasible design region, the algorithm creates three surrogate models. The algorithm repeats the process of sampling and updating the Pareto set, by assigning weighting factors to those surrogates in accordance with the values of the root mean squared error, until a Pareto frontier is discovered. AMSP was successfully employed to identify the Pareto set and the Pareto border. Utilizing multi-objective benchmark test functions and engineering design examples such as airfoil shape geometry of wind turbine, the unique approach was put to the test. The cost of computing the Pareto optima for test functions and real engineering design problem is reduced, and promising results were obtained.

Keywords: multi-objective optimization; mixed surrogates; Pareto frontier; wind turbine airfoil geometry



Citation: Younis, A.; Dong, Z. High-Fidelity Surrogate Based Multi-Objective Optimization Algorithm. *Algorithms* **2022**, *15*, 279. <https://doi.org/10.3390/a15080279>

Academic Editor: Angel A. Juan

Received: 25 June 2022

Accepted: 2 August 2022

Published: 7 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Numerous competing objectives and constraint functions, as well as time-consuming and expensive simulations, are present in the majority of real-world optimization issues. High-fidelity models must frequently be expensively modelled in order to address these problems. Finding the Pareto optimum of multi-objective optimization problems (MOO) necessitates evaluating expensive fitness and constraint functions. However, only a few tests are allowed in practice due to the restricted computing resources available to address such problems, notably black box functions. Researchers [1–3] found that the development of surrogate models employing a few high-fidelity solution assessments to replace computationally expensive models is promising and fruitful. Expensive functions are commonly replaced by surrogate models such as the radial basis function, Kriging, or response surface method [4].

In MOO problems demanding expensive computational analysis and simulation tools, for example FEA and CFD, surrogate models have grown in prominence and received more attention. Due to their capacity to imitate the real expensive model and their compliance with challenging computational requirements, surrogate models have been proved to be viable tools for MOO problems [5]. Numerous optimization applications demand labor- and resource-intensive fitness evaluations. The load considerably increases when dealing

with multiple-objective black box functions that need to be evaluated. The automated MOO algorithm (AMSP) proposed in this paper is based on surrogate approximations.

The suggested algorithm automatically constructs an ensemble of surrogates from the three available surrogate models based on a methodology that will be introduced and explained in detail in this paper. The algorithm adaptively adjusts the effective surrogate model to determine the Pareto frontier for the MOO problem. The Response Surface Function (RSF), Radial Basis Function (RBF), and Kriging model are employed as surrogate approximations in this work [6–11]. To fully make use of the potential design space, an efficient sampling technique must be used. Traditional sampling techniques are afflicted by the “curse of dimensionality”, which causes an exponential increase in the number of sampling points needed to explore the design space as the number of design variables rises. The Latin Hypercube Designs (LHD) method [12] was consequently modified in this work. LHD is renowned for randomly and evenly covering the practicable portion of the design space in its sampling. These significantly reduce computation costs and increase the probability of finding the Pareto frontier for MOO problems. The suggested algorithm (AMSP) consistently and accurately locates the Pareto frontier for MOO problems. The computational cost is significantly reduced when AMSP is used to locate the Pareto frontier for expensive black-box functions. AMSP was put to the test using benchmark test problems and actual, realistic engineering challenges in order to show off its performance. The outcomes are seen as promising and encouraging.

2. Literature Review

MOO has emerged as a possible approach to overcoming obstacles with conflicting goals. The most effective trade-offs between a set of criteria are often determined by using a group of Pareto optima to solve MOO issues. Finding the final Pareto frontier for use in practice is challenging, especially when there are several competing objectives to take into account.

In situations with two or three objective functions, it may be rather simple to describe the set of Pareto-optimal solutions in the objective function space. When we identify Pareto limits, we may completely understand the relationship between trade-offs between objectives. It has been demonstrated that surrogate models may accurately and efficiently identify Pareto-optimal solutions.

Effective and durable search strategies are greatly needed to tackle such challenging multi-objective optimization problems. The majority of algorithms are evolutionary ones such as GA, PSO, and others. Black box functions have only been shown for a select few. These are the solutions provided for multi-objective optimization problems that demand pricey analysis and simulation methods, such as multi-physics modeling and simulation, FEA, and CFD analyses. The optimization of computationally expensive black box functions needs more focus and should be addressed due to the challenges these functions have presented to the optimization community. A promising method that might help with the resolution of these problems is the use of surrogate models. The expensive model (function) is simulated and predicted using surrogate models, which have a reduced computational cost. MOOP with expensive high fidelity computational models, such as sophisticated simulations and thorough analyses, the standard multi-objective optimization methods, are challenging to perform directly due to the high computational cost of objective and constraint function evaluations. As a response to these problems, surrogate models gain in popularity.

There are typically two scenarios in which black box functions are used to address MOO problems. Numerous evolutionary algorithms (EAs) use the first scenario, which closely resembles the Pareto optimal [13–15]. Due to the costly evaluations of numerous non-Pareto set points required by these approaches, they are renowned for being computationally expensive. In the second scenario, surrogate models are used to assess each objective function. Naturally, the constructed surrogate models will determine how precise the Pareto-optimal border is.

Li et al. [16] created a hyper-ellipse surrogate to approach the Pareto optimum frontier for bicriteria convex optimization problems. If the established approximation model is not sufficiently accurate, the Pareto-optimal frontier will not be considered a good approximation of the genuine Pareto-optimal frontier. In order to make use of the multi-objective design region and find the Pareto set using approximation models, Wilson et al. [17] used two surrogate approximations (response surface and Kriging models). Prior to applying the optimization strategy, they employ a technique that minimizes the loss of surrogate approximation accuracy. Proos et al. [17] used the weighted and global criterion technique to construct an algorithm that incorporates several criterion optimizations into evolutionary structural optimization (ESO). Yang et al. [15] developed a strategy for managing surrogate models for MOO. The framework includes a sequentially-updated surrogate model and a GA-based method. Instead of before, as in the preceding case, the surrogate model is altered during the optimization process [14]. The accuracy of the surrogate model is crucial for maintaining the integrity of the identified border sets in their proposed method.

Yang's [15] research had a hard time locating frontier areas close to the extremes. The Pareto Set Pursuing (PSP) technique, proposed by Shan and Wang [18], entails creating sampling guidance functions based on surrogate models. It was said that PSP had enormous promise in terms of effectiveness, accuracy, and robustness. An artificial neural network (ANN) approximation was merged with NSGA-II by Nain and Deb [19] in order to achieve a computationally efficient search and enable the use of GAs on computationally expensive problems. A progressively updated design and analysis of computer experiments (DACE) model-based extended efficient global optimization method (ParEGO) for MOO challenges is presented by Knowles [20]. In order to develop a multi-objective design optimization approach, Kim and Chung [21] combined GA with Kriging. The developed method was tested in a wing platform design competition, proving its efficacy and viability. A novel multi-objective optimization approach based on the usage of surrogate models was developed by Liu et al. [22]. In each iteration, the approximation models are gradually produced by the response surface approximations. The Pareto optimum set proposed by the approximations is located using a multi-objective genetic algorithm. The proposed approach relies less on the accuracy of the surrogate models because it concentrates on identifying the real Pareto ideal. Lim in [23] explored the use of substitute models in evolutionary search. Jang in [24] employed an adaptive approximation framework to address the processing cost of the entire stochastic fatigue analysis in the optimization process.

An adaptive Kriging model was created by Yang et al. [25] that increases the accuracy of the approximations by adding additional points to the model with each iteration. Zhao et al. [26] suggested a dynamic Kriging methodology, which uses several sets of approximation functions in various groups of points, was proposed as a way to regulate the nonlinearity of the model space [26]. The suggested technique has proven to be extremely successful when applied to challenging optimization problems [27]. By automatically choosing pertinent surrogates during the search process, Gu et al. [28] developed an optimization approach to increase search efficiency by integrating a dynamic Radial Basic Function (RBF) with an adaptive sampling technique. Diez et al. [29] developed a method for improving both the existing solution and the approximation model. A dynamic RBF function-based strategy was developed by Volpi et al. [30] who claimed it was successful in solving high-dimensional problems. Iuliano [31] examined and discussed a number of adaptation strategies.

The proposed algorithm is designed to handle highly non-linear black-box multi-objective optimization problems. The algorithm is unique because it takes advantage of three surrogates to identify the Pareto set and the Pareto frontier with less computational cost. AMSP adaptively combines three surrogate models based on a selection criterion that will be discussed later. To find the Pareto frontier for the multi-objective problem, the method automatically assigns a weight factor to each surrogate model based on calculated RMSE values. LHD was utilized to generate sampling points to refine the search for optimal Pareto sets. AMSP uses the ensemble of surrogates to identify the Pareto frontier. These

factors aided in the decrease of computation costs and boosted the likelihood of identifying the Pareto frontier for MOO problems.

3. Multi-Objective Optimization

MOO is defined as a set of design variables that satisfy constraints and optimize a vector function whose constituents are the objective functions. It can be expressed as follows:

Optimize the vector function

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \dots, f_k(\bar{x})]^T \quad (1)$$

by determining the decision variables vector;

$$\bar{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T \quad (2)$$

which will satisfy the m inequality constraints

$$g_i(\bar{x}) \geq 0, \quad i = 1, 2, \dots, m \quad (3)$$

and the p equality constraints

$$h_j(\bar{x}) = 0, \quad j = 1, 2, \dots, p \quad (4)$$

where $x = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables. In other words, we wish to determine from among the set f of all numbers, which satisfy Equations (3) and (4). In particular set $x_1^*, x_2^* \dots x_k^*$ which yields the optimum values of all the objective functions.

Pareto Frontier

The Pareto frontier or Pareto set is the set of all Pareto-efficient results. If no feasible vector x exists that decreases some objective functions while simultaneously increasing at least one other objective function, then the vector of, x^* is Pareto-optimal. Following are some mathematical ways to express the Pareto optimality:

A vector of x^* is a Pareto optimum if and only if, for any x and i ,

$$f_j(x) \leq f_j(x^*), \quad j = 1, \dots, m; \quad j \neq i, \quad f_i(x) \geq f_i(x^*) \quad (5)$$

Finding Pareto set points allows for the discovery of a Pareto frontier, which is the main purpose of multi-objective optimization (MOO). Claiming that these points are in the Pareto set or the Pareto frontier is challenging unless there is a strong method to determine whether they are in the set or not. This could be quickly ascertained if a fitness function is customized. The fitness function, which was utilized in this paper, will be discussed in the section that follows.

4. The Proposed Algorithm

The proposed algorithm's main objective is to identify the Pareto frontier with the least amount of computing time for expensive black-box functions. This goal can be achieved if appropriate tools, such as effective sampling techniques, are used. Because it provides equal and random samples in the design space, the LHD sampling technique is used in this work. The Pareto frontier should be approached repeatedly whenever more samples are generated. A sampling guidance function is needed to judge if these sample sites are near to or on the Pareto frontier. The following section introduces the sampling guidance procedure.

4.1. Sampling Procedure

There are many statistical methods available today for creating sample points based on a specific probability density function (PDF). Latin Hypercube Designs (LHD) sampling technique was used to create random and evenly distributed sample points in the design space of interest. The search process starts with the creation of a surrogate approximation model from a small number of sample points, then generates many points using the model. Finally, the points are sorted, and a cumulative function that is similar to the cumulative density function (CDF) proposed in [32] is created by adding up all the function values. The generation of sample points based on a specific probability density function can be done statistically using a variety of methods nowadays (PDF). Latin Hypercube Designs (LHD) sampling technique was used to create random and evenly distributed sample points in the design space of interest. The search starts by building a surrogate model from a small number of sample points, then generates many points using the surrogate model, sorts the points, and then builds a cumulative function similar to the cumulative density function (CDF) proposed in [32] by adding up all the function values.

4.2. Surrogate Models

In this study, surrogate models served as crucial research tools. Three popular surrogate approximations were used: Kriging, RBF, and RSM. The next sections present each surrogate along with its mathematical formulations.

4.2.1. Kriging

The Kriging model is one of the most well known spatial interpolation models for substituting the numerical relationship between input and output variables. It loses a lot of efficiency when dealing with significant design variances. On the other hand, Kriging surrogate models can significantly reduce the computational cost of regression because they need significantly fewer sample data to be generated. Additionally, more trustworthy prediction results can be produced because the Kriging fitting technique concentrates on more representative samples.

The Kriging model regards the function of interest as a realized random function (stochastic process). As a result, a linear combination of a global model and deviations, Equation (6), is presented as the Kriging mathematical model:

$$y(x) = f(x) + Z(x) \quad (6)$$

where $y(x)$ is the unknown deterministic response, $f(x)$ is a known (usually polynomial) function of x , and $Z(x)$ is a realization of a stochastic process with mean zero, variance σ^2 , and non-zero covariance.

4.2.2. Radial Basis Function (RBF)

The interpolation of scatter data using radial basis functions has been shown to be highly accurate in high-dimensional problems. Radial basis function interpolation, Equation (7), is used to approximate the form.

$$\hat{f}(x) = \sum_{n=1}^N \omega_n \varphi(x - x_n) \quad (7)$$

where $\varphi(x) = \varphi(\|x\|)$ is a radial function. The positions x_n , $n = 1 \dots N$ are known as the RBF centers.

4.2.3. Response Surface Function (RSF)

RSF was initially created to model experimental results [6], after which it was expanded to include numerical experiment modeling. RSF is used in design optimization to reduce

the cost of pricey analytical methods (for example, FEA or CFD analyses) as well as the related numerical noise. Equation (8) can be used to express the response function:

$$y = f(x_1, \dots, x_n) + \varepsilon \tag{8}$$

where ε represents the noise or the error observed in the response y . The surface represented by $f(x_1, \dots, x_n)$ is called a response surface.

4.3. Ensemble of Surrogate Models

The three surrogate models, RSF, RBF, and KRG, are integrated in an appropriate linear approach to produce a better-weighted ensemble of surrogates that can duplicate the high-fidelity surrogate in the feasible region of the design space.

Equation (9) represents the ensemble of surrogate’s equation:

$$\hat{y}_{ensemble}(x) = \beta_q RSF(x) + \beta_r RBF(x) + \beta_k KRG(x) \tag{9}$$

where

$$\beta_q + \beta_r + \beta_k = 1 \tag{10}$$

where $\hat{y}_{ensemble}(x)$ is the surrogate model to the analysis/simulation function, $f(X)$, at sample point x ; $RSF(x)$, $RBF(x)$ and $KRG(x)$ represents the *RSF*, *RBF* and *KRG* surrogate models respectively; β_q , β_r and β_k are weighted coefficients that control how much each model contributes to the combined surrogate.

4.4. Automated Ensemble of Surrogate’s Selection

Three surrogate models are built after producing a few sample points in the space of interest using Latin Hypercube Designs. RSF, RBF and Kriging. These surrogates’ root mean square error (RMSE) is calculated. In the next step, the surrogate with the highest RMSE value (close to 1) is chosen to be heavily depended on and assigned high weight factor and the one with low RMSE (close to zero) is assigned the lowest weight factor. Kriging typically yields lower RMSE than RSF and RBF. In this method, a rule was established to guide the mixed surrogate construction process, which is to first assess the RMSE and then automatically assign weight factor based on the value of the RMSE. The weight factor is selected as percentage so that the surrogate with RMSE close to 1 gets a weight factor 60% and the one with low RMSE gets a weight factor of 10%. The surrogate that yields RMSE in between gets 30% weight factor. If RMSE is close to 1, build Kriging. If the contrary is true, then build either RSF or RBF based on which one has RMSE closer to 1.

4.5. Maximum Fitness Function

There are numerous approaches to define the fitness function for an objective function problem. The function that comes next is known as the maximal fitness function. Equation (11) identifies the fitness function used in this study. Numerous programs have successfully exploited this capability. Equation (12) illustrates the modified maximum fitness function that was used in this study.

$$P(x_i) = \underbrace{\max}_{j \neq i; j \in P} [\underbrace{\min}_{1 \leq s \leq k} \{ f_s(x_i) - f_s(x_j) \}] \tag{11}$$

$$P_i = \max \left(\min \left(f_{s1}^i - f_{s1}^j, f_{s1}^i - f_{s1}^j, \dots, f_{sm}^i - f_{sm}^j \right) \right) \tag{12}$$

where, P_i denotes the fitness value of the i th design; f_{sk}^i is scaled to k th objective function value of the i th design, $k = 1, \dots, m$. The max in Equation (12) is over all other designs

$j \neq i$ in the set and the min is over all the objectives. The objectives $f_{s1}, f_{s2}, \dots, f_{sm}$ in Equation (13) are scaled to a range [0, 1]. For example, for f_{s1}^i ,

$$f_{s1}^i = \frac{unscf_{1,i} - unscf_{1,min}}{unscf_{1,max} - unscf_{1,min}} \quad (13)$$

where, $unscf_{1,i}$ denotes the un-scaled value of the first objective for the i th design; $unscf_{1,min}$ denotes the maximum f_{s1}^i un-scaled value of the first objective among all designs; and $unscf_{1,max}$ denotes the minimum un-scaled value of the first objective among all designs. In case that an objective function is a constant, the scaled objective function value is taken as 1 in this work.

4.6. The Proposed Algorithms Steps:

The steps of the suggested approach are summarized and discussed in the steps below:

1. Sampling initial random design points. Creating a small number of randomly distributed sample points in the design space. The sample procedure is carried out using LHD [12].
2. The black box function (expensive function) is used to detect the current frontier points after evaluating the generated sample points.
3. Calculating the built surrogates' root mean square error (RMSE). In this step, three mixed surrogates are introduced to imitate and replace the expensive function: the response surface function (RSF), radial basis function (RBF), and Kriging (KRG) surrogate model. After these substitutes are fitted to the previously acquired sample points, RMSE is calculated. The surrogate with a higher RMSE value (near to 1) is automatically adjusted by the algorithm, and it is given a higher weight factor than the other surrogates with lower RMSE values (close to 0). The surrogate with the lowest RMSE values receives a weight percentage of 10%, the surrogate with the highest RMSE values receives a weight percentage of 60%, and the surrogate with all other RMSE values receives a weight percentage of 20%. The subsequent stages should make use of the selected mixed surrogate.
4. Creating a lot of cheap points in a short amount of time. They are used to assess the enormous number of sample points produced by LHD after the development of the mixed surrogate (cheap function).
5. Combining the sample points. The preliminary approximated Pareto border is identified or moved towards in this stage by combining all generated sample points (expensive and cheap locations).
6. Identifying the candidate points out of all the ones that already exist (points obtained in the previous step). These points are very likely to become Pareto set points.
7. For fresh sample points, evaluate fitness functions. The expensive black-box functions were used to evaluate the points collected in step 5.
8. To identify the new frontier set, new sample points are combined with expensive frontier points.
9. If convergence criteria (termination criteria) are met, the method ends; otherwise, return to step 4.

4.7. Termination Criterion

Two termination criteria were adapted in this work.

1. If the difference between frontier points after two consecutive iterations is less than 0.0001, then terminate: and
2. If the maximum number of specified number of iterations has been reached, then the algorithm should terminate.

Figure 1 depicts the proposed approach's flow diagram. It explains how the algorithm works in a nutshell.

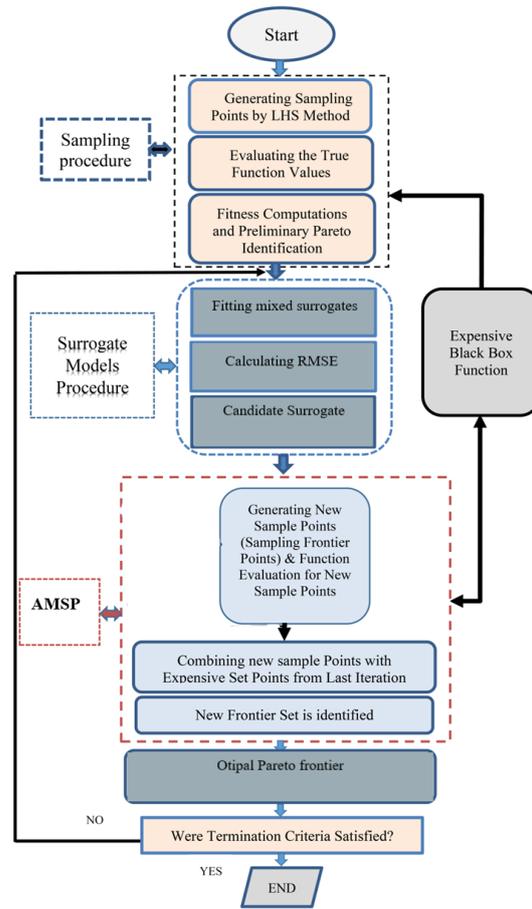


Figure 1. Flowchart of the Proposed approach (AMSP).

5. Numerical Examples and Results

The suggested approach is tested and validated using a number of well-known multi-objective optimization benchmark test functions from the literature, which are reported in Table 1. The results of the four test issues are shown in this section. The results of optimizing a real-world example are also provided in this section.

5.1. Test Functions

The test functions tested using the proposed approach are reported in Table 1.

Table 1. Multi-objective test functions.

	Test Function	Constraints	Domain
F1	Minimize $f_1(x) = (x_1^2 - 5)^2 + (10x_2/A - 6)^2$; Minimize $f_2(x) = (x_1^2 - 7)^2 + (10x_2/A - 6)^2$		$x_1, x_2 \in [-10, 10]$ Where $A = 10$
F2	Minimize $f_1(x) = (x_1)^5 + x_2$; Minimize $f_2(x) = (x_1)^5 + 1 - x_2$		$x_1, x_2 \in [0, 1]$
T1	Minimize $f_1(x) = (x_1 - 2)^2 + (x_1 - 1)^2$; Minimize $f_2(x) = x_1^2 + (x_2 - 6)^2$	$h_1(x) = x_1 - 1.6 \leq 0$; $h_2(x) = 0.4 - x_1 \leq 0$; $h_3(x) = x_2 - 5 \leq 0$; $h_4(x) = 2 - x_2 \leq 0$	$x_1 \in [0.4, 1.6]$ $x_2 \in [2, 5]$
T2	Minimize $f_1(x) = (x_1 + x_2 - 7.5)^2 + (x_2 - x_1 + 31)^2/4$; Minimize $f_2(x) = (x_1 - 1)^2/4 + (x_2 - 4)^2/2$	$h_1(x) = 2.5 - (x_1 - 2)^3/(2 - x_2) \leq 0$; $h_2(x) = 3.85 + 8(x_2 - x_1 + 0.65)^2 - x_2 - x_1 \leq 0$	$x_1 \in [0, 5]$ $x_2 \in [0, 3]$
T3	Minimize $f_1(x) = 25 - (x_1^3 + x_1^2(1 + x_2 + x_3) + x_2^3 + x_3^3)/10$; Minimize $f_2(x) = 35 - (x_1^3 + 2x_2^2 + x_2^2(2 + x_1 + x_3) + x_3^3)/10$; Minimize $f_3(x) = 50 - (x_1^3 + x_2^3 + 3x_3^3 + x_3^2(3 + x_1 + x_3))/10$;	$h_1(x) = 12 - x_1 - 2x_2 - x_3 \geq 0$	$x_1, x_2, x_3 \in [0, 5]$

5.2. Wind Turbine Airfoil Geometry Optimization

The geometry of a typical GT compressor blade airfoil [33] is optimized in this example using the suggested algorithm. In both design and off-design scenarios, the objective is to lower the total pressure loss coefficient. NURBS curves were used to parameterize the geometry input used in the optimization process. This method treats the positions of the non-uniform rational basis spline (NURBS) control points as design variables. Four NURBS curves for each of the four segments of the compressor airfoil, each having nine control points, make up the airfoil. Because each control point has two coordinates, x and y, there are a total of 72 design variables. The junction points of the segments are one of 16 known and set parameters. The remaining 16 parameters are found by enforcing C2-continuity (which requires the second derivative to be zero at the endpoints) at the intersections of the segments. As a result, 40 variables will remain as design variables in the optimization procedure. However, to limit the number of design factors while maintaining a high level of geometric flexibility, the geometry of the leading edge (LE) and trailing edge (TE) is maintained constant to save CPU time.

The results of the airfoil optimization procedure strongly depend on the formulation of the objective function. In order to simulate 2-D fluid flow, the geometry code creates parameterized profiles that are then imported into the computational fluid dynamics program COMSOL CFD. The acceptable profiles' post-processed outputs are then fed into the fitness calculation portion, where the airfoil loss values, L , should be minimized with respect to any shape. The single objective function is shown in Figure 2 as follows:

$$\begin{aligned} \text{Minimize } L_1\% &= (a_1 L_s) * 100 \\ \text{Minimize } L_2\% &= (a_2 L_d) * 100 \\ \text{Minimize } L_3\% &= (a_3 L_c) * 100 \end{aligned} \tag{14}$$

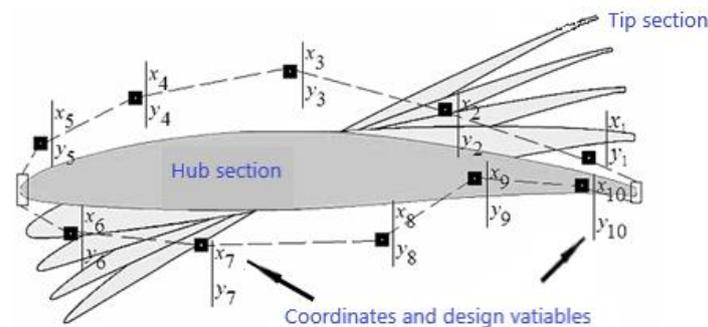


Figure 2. Schematic diagram of airfoil shape with twenty shape design variables.

Subject to:

$$\begin{aligned} (|y_3 - y_7| \text{ and } |y_3 - y_8|) &\geq 15\% \text{ of the chord} \\ 0 &\leq x_i / \text{chord} \leq 100\% \\ 40\% &\leq y_i / \text{chord} \leq 55\% \end{aligned} \tag{15}$$

where a_i are weighting factors. The reduction of total pressure loss on the right (L_c) and left (L_s) sides of the design point significantly expands the operating range. The quantities y_i and x_i are design variables as shown in Figure 2.

The following are the weighting factors for the optimization process: $a_1 = 0.20$, $a_2 = 0.70$, and $a_3 = 0.10$.

$$\begin{aligned} L_s &= (P_{o1} - P_{o2})_{stall} / 0.5\rho V_1^2, & L_d &= (P_{o1} - P_{o2})_{des} / 0.5\rho V_1^2, \\ L_c &= (P_{o1} - P_{o2})_{choke} / 0.5\rho V_1^2, \end{aligned} \tag{16}$$

where P_{o1} is intake total pressure, P_{o2} is outlet total pressure, and V_1 is inlet velocity.

6. Results and Discussion

Multiple benchmark multi-objective optimization problems and a real-world engineering design challenge were used to test the proposed technique. Promising outcomes were gathered using the suggested algorithm. There were numerous computational optimization runs (approximately five runs). As noted earlier, Table 2 presents samples of the outcomes from five separate runs. The actual, real-world example (geometry of the airfoil in a wind turbine) required between 986 and 1172 iterations, with 242 serving as the median. The median number of function evaluations is 1213, with a range of 16,582 to 19,934. RBF surrogate model is the winner in the majority of test runs. The median number of convergent Pareto points, which varies from 729 to 793, is 760.

Table 2. Test results of the introduced algorithm on multi-objective test functions.

F1						F2					
Run	Ite.	Eval	#Par.P	Surr.	RMSE	Run	Ite.	Eval	Par.P	Suur.	RMSE
1	37	75	48	RBF	0.9901	1	24	246	137	RBF	0.9721
2	32	61	45	RSF	0.9899	2	22	281	188	KRG	0.9704
3	23	56	35	KRG	0.9879	3	23	275	167	RBF	0.9569
4	30	66	43	RBF	0.9744	4	21	458	309	RSF	0.9565
5	33	71	47	RSF	0.9904	5	20	410	357	RBF	0.9777
T1						T2					
Run	Ite.	Eval	#Par.P	Surr.	RMSE	Run	Ite.	Eval	Par.P	Suur.	RMSE
1	19	446	363	RBF	0.9812	1	27	58	39	RSF	0.9744
2	19	443	358	RSF	0.9963	2	29	59	37	RBF	0.9730
3	16	368	306	KRG	0.9811	3	34	73	44	KRG	0.9773
4	17	379	313	RSF	0.9792	4	28	65	44	RSF	0.9302
5	15	332	264	RBF	0.9971	5	38	87	46	RBF	0.9565
T3						P1					
Run	Ite.	Eval	#Par.P	Surr.	RMSE	Run	Ite.	Eval	Par. P	Suur.	RMSE
1	250	1216	760	RSF	0.9325	1	1172	18,857	3464	RSF	0.9772
2	234	1213	765	RBF	0.9355	2	1218	19,934	3615	RSF	0.9821
3	273	1294	793	RBF	0.9781	3	986	16,582	3146	RSF	0.9763
4	232	1170	752	RSF	0.9932	4	1019	19,121	3623	RBF	0.9775
5	219	1172	729	RBF	0.9946	5	997	18,587	3455	RBF	0.9801

Par. P = Number of Pareto points.

A comparison of the complete airfoil geometry was carried out, as seen in Figure 3. Solid lines depict the geometry of the initial (Datum) unoptimized airfoil. The unoptimized airfoil geometry is shown by the continuous line, while the airfoil shape generated by the AMSP optimizer is shown by the dashed line. The most obvious inference from this graph is that the geometry of the airfoil in the second half of the chord, from maximum thickness to TE, has been significantly affected by the optimization process. An enhanced airfoil shape is produced when three surrogates are combined into an ensemble. This indicates better pressure distribution and greatly increases the lifting force and lowers the drag force of the turbomachinery blades.

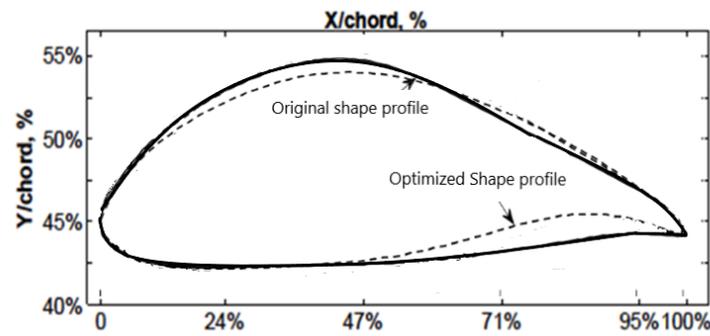


Figure 3. Airfoil shape design before and after optimization process.

Figures 4–7 present the performance space and the evaluated points for test functions F1, F2, T1, T2, and T3 respectively. Figure 4a displays the sampling points coordinates in the feasible design space, Figure 4b shows sampling design at which the sampling design were evaluated using the objective functions. Figure 4c displays the Pareto design coordinates and Figure 4d displays the Pareto frontier after all Pareto coordinates were converged iteration after iteration to form the Pareto frontier. The same can be seen in Figures 5–8 for test functions F1, F2, T1, T2, and T3. It is quite noticeable that the proposed algorithm successfully converged to optimal Pareto set and successfully identified the Pareto frontier for all tested benchmark multi-objective functions and most importantly for the practical problem, which is the wind turbine blade shape/profile optimization. The algorithm was capable of identifying the Pareto frontier for a real-life practical problem, seen in Figure 8, with three objective functions and 40 design variables. By identifying the optimal design variables $x_1, y_1, \dots, x_{10}, y_{10}$, the optimal airfoil profile was generated that minimizes the loss values L_1, L_2 , and L_3 .

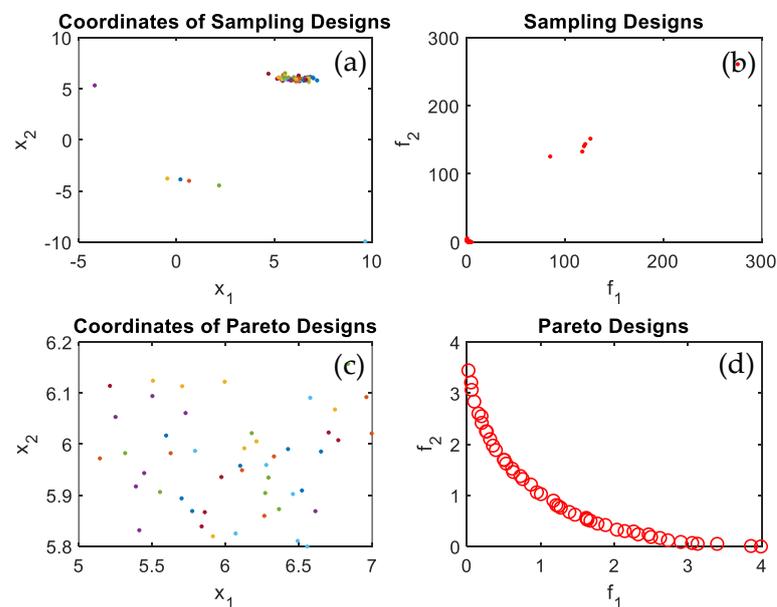


Figure 4. Computational simulations Pareto frontier determination of test function F1 (a) coordinates of sampling points; (b) sampling points in feasible design space; (c) coordinates of Pareto points; (d) the seeking Pareto frontier.

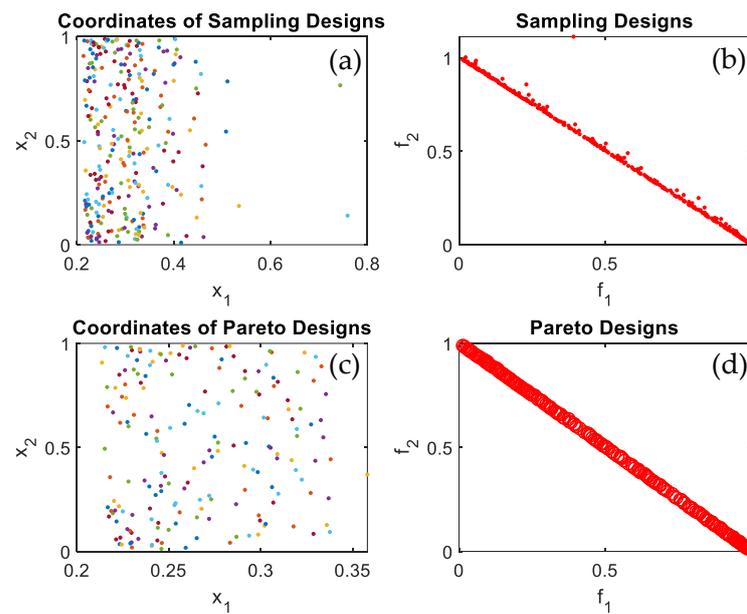


Figure 5. Computational simulations Pareto frontier determination of test function F2 (a) coordinates of sampling points; (b) sampling points in feasible design space; (c) coordinates of Pareto points; (d) the seeking Pareto frontier.

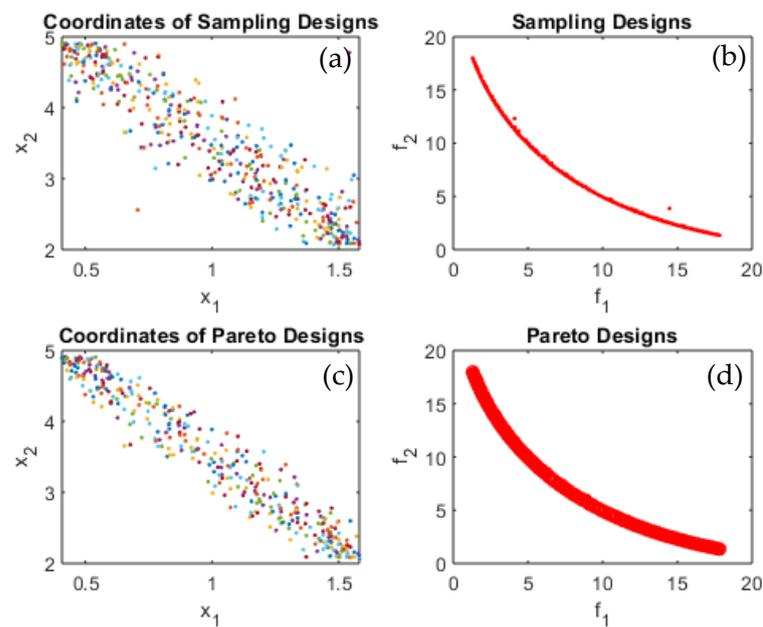


Figure 6. Computational simulations Pareto frontier determination of test function T1 (a) coordinates of sampling points; (b) sampling points in feasible design space; (c) coordinates of Pareto points; (d) the seeking Pareto frontier.

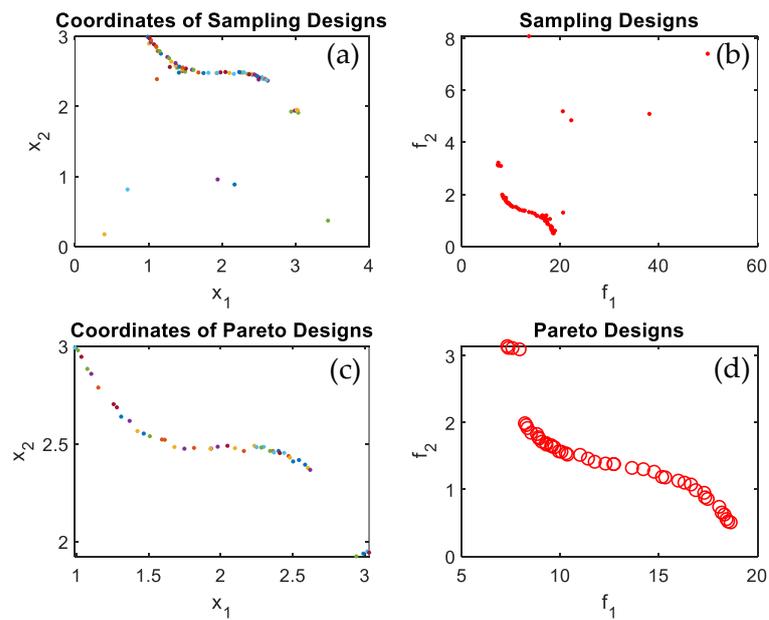


Figure 7. Computational simulations Pareto frontier determination of test function T2 (a) coordinates of sampling points; (b) sampling points in feasible design space; (c) coordinates of Pareto points; (d) the seeking Pareto frontier.

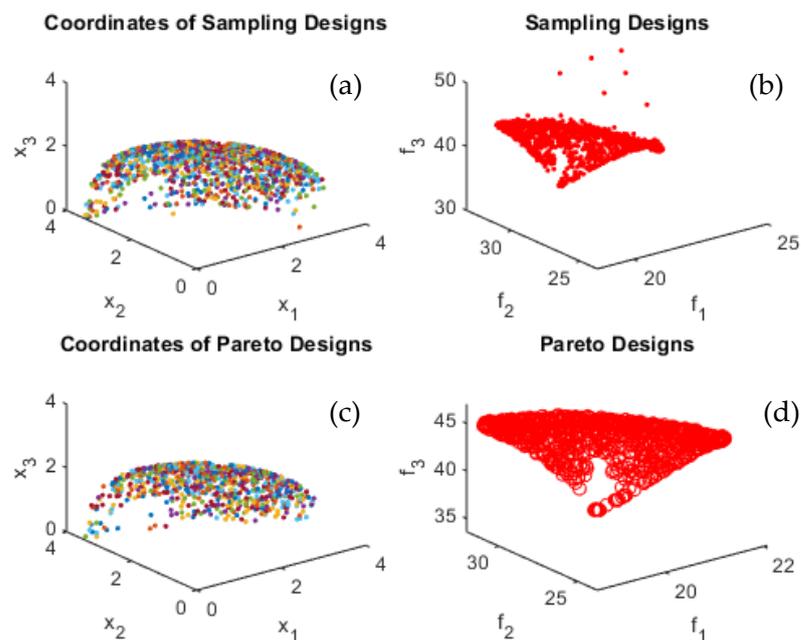


Figure 8. Computational simulations Pareto frontier determination of test function T3 (a) coordinates of sampling points; (b) sampling points in feasible design space; (c) coordinates of Pareto points; (d) the seeking Pareto frontier.

Table 3 reports the total number of iterations, the total number of objective function evaluations, and the total number of Pareto points in the ideal Pareto frontier. The efficient Pareto set was successfully identified with a reasonable number of Pareto set points and a low computing cost using RBF, which is devoted to the sample points, and Kriging, which interpolates the sample points and gives good accuracy. Table 3 displays the variation range, median, and Pareto set points for the iterations, function evaluations, and number of iterations.

Table 3. Summary of test results of AMSP.

Test Function	Number of Iterations		Number of Evaluations		Number of Pareto Set Points	
	Range	Median	Range	Median	Range	Median
F1	[23–37]	31	[56–75]	66	[35–48]	44
F2	[20–24]	22	[246–458]	334	[137–357]	232
T1	[15–23]	18	[332–446]	394	[264–363]	321
T2	[27–38]	32	[58–87]	69	[37–46]	42
P1	[219–273]	242	[1170–1294]	1213	[729–793]	760

For each test problem, more than five runs have been completed because LHD creates sample points in the region of interest at random. The best outcome from those several runs is represented by these results. For five distinct runs, the quantity of iterations, total quantity of evaluated points, and quantity of converged frontier points were noted.

It worth mentioning that the task can be completed in a relatively small number of iterations. All the evaluated points are plotted together with the feasible performance space, as shown in Figures 4–8 to better observe the accuracy of converged frontier points. The converged frontier points are extremely close to the genuine Pareto frontier, as can be seen by the graphs.

Benchmark test functions were used to compare the proposed algorithm’s performance to that of other MOO algorithms. Table 4 shows that AMSP outperforms the other MOO techniques, such as Pareto Set Pursuing (PSP) and Multi-objective Optimization Genetic Algorithm (MOGA). In terms of the number of fitness evaluations, AMSP outperformed the other algorithms in terms of efforts and resources needed. In addition, the cost of computation was lowered reflected in reduction of number of function evaluations and computation time. Furthermore, when AMSP was utilized, more Pareto set points were obtained. As a result, a more accurate and smoother Pareto border was identified.

Table 4. Performance comparison of multi-objective optimization algorithms.

Test Problem		Number of Iterations		Number of Fun. Evaluations		Number of Pareto Set Points	
		Range	Median	Range	Median	Range	Median
F1	AMSP	[23–37]	31	[56–75]	66	[35–48]	44
	PSP	[21–35]	33	[58–79]	68	[36–44]	42
	MOGA	[520–710]	480	[919–1087]	995	[20–24]	22
F2	AMSP	[20–24]	22	[246–458]	334	[137–357]	232
	PSP	[22–24]	23	[260–440]	339	[151–360]	241
	MOGA	[492–780]	503	[1060–1313]	1197	[112–126]	119
T1	AMSP	[15–23]	18	[332–446]	394	[264–363]	321
	PSP	[17–23]	20	[356–476]	400	[33–42]	37
	MOGA	[310–316]	313	[16,246–20,174]	18,841	[19–22]	21
T2	AMSP	[27–38]	32	[58–87]	69	[37–46]	42
	PSP	[52–88]	70	[55–83]	71	[29–44]	35
	MOGA	[418–489]	450	[19,900–22,446]	21,000	[17–21]	19
P1	AMSP	[219–273]	242	[1170–1294]	1213	[729–793]	760
	PSP	[240–264]	246	[1210–1290]	1246	[604–616]	606
	MOGA	[620–980]	800	[310,200–360,900]	33,555	[14–18]	16

AMSP = Automated Multiobjective Surrogate Based Pareto Finder; PSP = Pareto Set Pursuing; MOGA = Multiobjective Genetic Algorithm

The reduction in computation cost is reflected in a smaller number of fitness function evaluations, as seen in Figure 9. In terms of the number of fitness tests required to determine the Pareto optima, AMSP is slightly better than PSP but far better than MOGA. This suggests that AMSP is suitable for complex optimization problems and black box functions, as well as practical engineering applications, due to its ability to handle the burden of effectively

exploiting the design space and simulating expensive fitness functions with inexpensive surrogates that can be easily evaluated with fewer resources.

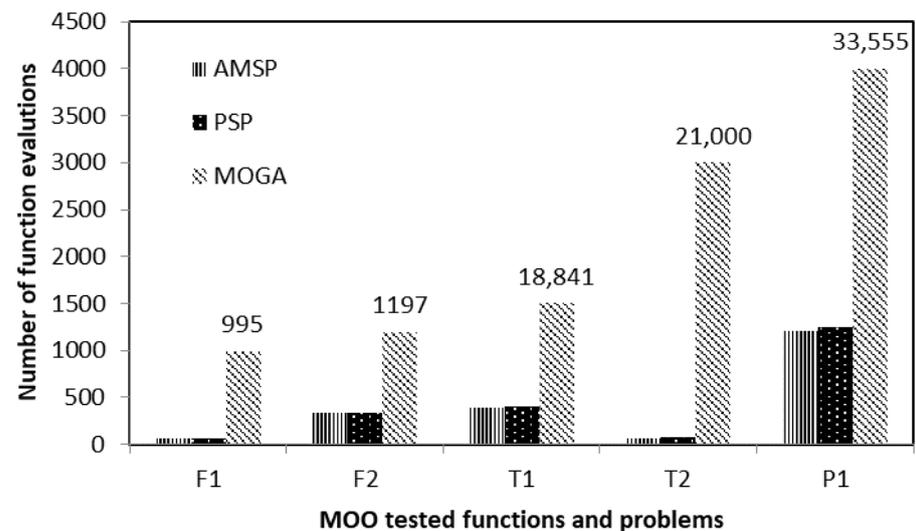


Figure 9. Performance comparison of used optimization algorithms on test problems in terms of computational cost.

7. Conclusions

A novel and new multi-objective optimization algorithm for identifying the Pareto frontier has been proposed. To find the best Pareto frontier for expensive and black box functions, the suggested algorithm AMSP automatically selects and fits the suitable surrogate approximation model based on RMSE values. Without prior knowledge of the target function, AMSP provides decision makers with a Pareto set for choices. Even if the frontier surface is highly nonlinear or discontinuous, AMSP algorithm can produce solutions that reflect the full Pareto-optimal frontier. With fewer efforts, resources, and expensive function evaluations, AMSP can identify the best Pareto frontier for all tested benchmark test functions with less computational cost and resources. To test the suggested algorithm and show its benefits and drawbacks, a variety of multi-objective benchmark test problems and a practical engineering multi-objective application were examined. The obtained results demonstrated that AMSP could identify the Pareto frontier with comparably good accuracy. When compared against other MOO algorithms, the newly suggested algorithm produced promising outcomes.

Author Contributions: Conceptualization, A.Y.; Data curation, Z.D.; Formal analysis, A.Y.; Investigation, A.Y.; Methodology, A.Y.; Resources, Z.D.; Software, A.Y.; Writing—original draft, A.Y.; Writing—review & editing, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cassioli, A.; Schoen, F. Global optimization of expensive black box problems with a known lower bound. *J. Glob. Optim.* **2013**, *57*, 177–190. [[CrossRef](#)]
2. Jin, Y. Surrogate-assisted evolutionary computation: Recent advances and future challenge. *Swarm Evol. Comput.* **2011**, *1*, 61–70. [[CrossRef](#)]

3. Ponweiser, W.; Wagner, T.; Biermann, D.; Vincze, M. Multi-objective Optimization on a Limited Budget of Evaluations Using Model-Assisted S-Metric Selection. In *Parallel Problem Solving from Nature—PPSN X*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 784–794.
4. Younis, A.; Dong, Z. Trends, features, and tests of common and recently introduced global optimization methods. *Eng. Optim.* **2010**, *42*, 691–718. [[CrossRef](#)]
5. Younis, A.; Dong, Z. Metamodel multi-objective optimization tool for mechatronic system design. In Proceedings of the 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Suzhou, China, 8–10 July 2012; pp. 218–223.
6. Box, G.E.P.; Wilson, K.B. On the Experimental Attainment of Optimum Conditions. *J. R. Stat. Soc. Ser. B* **1995**, *13*, 1–45.
7. Hardy, R.L. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **1971**, *76*, 1905–1915. [[CrossRef](#)]
8. Boomhead, D.S.; Lowe, D. *Radial Basis Functions, Multi-Variable Functional Interpolation, and Adaptive Networks*; Technical Report; Royal Signals and Radar Establishment Malvern (RSRE): Worcestershire, England, 1988; p. 4148.
9. Broomhead, D.S.; Lowe, D. Multivariable functional interpolation and adaptive networks (PDF). *Complex Syst.* **1988**, *2*, 321–355.
10. Cressie, N. Spatial prediction and ordinary Kriging. *Math. Geol.* **1988**, *20*, 405–421. [[CrossRef](#)]
11. Mckay, M.D.; Bechman, R.J.; Conover, W.J. A comparison of three methods for selecting values of input variables in the analysis techniques for computer codes. *Technometrics* **1979**, *21*, 239–245.
12. Tang, B. Orthogonal array based Latin hypercubes. *J. Am. Stat. Assoc.* **1993**, *88*, 1392–1397. [[CrossRef](#)]
13. Tappeta, R.V.; Renaud, J.E. Interactive Multi-objective Optimization Design Strategy for Decision Based Design. *ASME J. Mech. Des.* **2001**, *123*, 205–215. [[CrossRef](#)]
14. Wilson, B.; Cappelleri, D.; Simpson, T.W.; Frecker, M. Efficient Pareto frontier exploration using surrogate approximations. *Optim. Eng.* **2001**, *2*, 31–50. [[CrossRef](#)]
15. Yang, B.; Yeun, Y.S.; Ruy, W.S. Managing approximation models in multi-objective optimization. *Struct. Multidisc. Optim.* **2002**, *24*, 141–156. [[CrossRef](#)]
16. Li, Y.; Fadel, G.M.; Wiecek, M.M. Approximating Pareto curves using the hyperellipse. In Proceedings of the 7th AIAA/USAF/NASAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, USA, 2–4 September 1998.
17. Proos, K.A.; Steven, G.P.; Querin, O.; Xie, Y.M. Metacriterion Evolutionary Structural Optimization Using the Weighted and the Global Criterion Methods. *AIAA J.* **2001**, *399*, 2006–2012. [[CrossRef](#)]
18. Shan, S.; Wang, G.G. Ann Efficient Pareto Set Identification Approach for Multi-objective Optimization on Black-Box Functions. *ASME* **2005**, *1227*, 866–874.
19. Nain, P.K.S.; Deb, K. A Computationally Effective Multi-Objective Search and Optimization Techniques Using Coarse-to-Fine Grain Modeling. In Proceedings of the 2002 PPSN workshop on Evolutionary Multiobjective Optimization, Granada, Spain, 7–11 September 2002.
20. Knowles, J. ParEGO: A Hybrid Algorithm with On-Line Landscape Approximation for Expensive Multi-objective Optimization Problems. *IEEE Trans. Evol. Comput.* **2005**, *10*, 50–667. [[CrossRef](#)]
21. Kim, S.; Chung, H.-S. Multi-objective Optimization Using Adjoint Gradient Enhanced Approximation Models for Genetic Algorithms. In Proceedings of the ICCSA 2006, LNNCS 3984, Glasgow, UK, 8–11 May 2006; pp. 491–502.
22. Liu, G.P.; Han, X.; Jiang, C. A novel multi-objective optimization method based on an approximation model management technique. *Comput. Methods Appl. Mech. Eng.* **2008**, *197*, 2719–2731. [[CrossRef](#)]
23. Lim, D.; Jin, Y.; Ong, Y.-S.; Sendhoff, B. Generalizing surrogate-assisted evolutionary computation. *IEEE Trans. Evol. Comput.* **2010**, *14*, 329–355. [[CrossRef](#)]
24. Jang, B.S.; Ko, D.E.; Suh, Y.S.; Yang, Y.S. Adaptive approximation in multi-objective optimization for full stochastic fatigue design problem. *Mar. Struct.* **2009**, *22*, 610–632. [[CrossRef](#)]
25. Yang, Q.; Huang, J.; Wang, G.; Karimi, H.R. An adaptive metamodel-based optimization approach for vehicle suspension system design. *Math. Probl. Eng.* **2014**, *2014*, 965157. [[CrossRef](#)]
26. Zhao, L.; Choi, K.; Lee, I.; Gorsich, D. A metamodel method using dynamic kriging and sequential sampling. In Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX, USA, 13–15 September 2010.
27. Lee, I.; Choi, K.; Zhao, L. Sampling-based RBDO using the dynamic kriging (d-kriging) method and stochastic sensitivity analysis. *Struct. Multidiscip. Optim.* **2011**, *44*, 299–317. [[CrossRef](#)]
28. Gu, J.; Li, G.Y.; Dong, Z. Hybrid and adaptive meta-model-based global optimization. *Eng. Optim.* **2012**, *44*, 87–104. [[CrossRef](#)]
29. Diez, M.; Volpi, S.; Serani, A.; Stern, F.; Campana, E.F. Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences. In *Computational Methods in Applied Sciences*; Springer: Cham, Switzerland, 2019; pp. 213–228.
30. Volpi, S.; Diez, M.; Gaul, N.J.; Song, H.; Iemma, U.; Choi, K.K.; Campana, E.F.; Stern, F. Development and validation of a dynamic metamodel based on stochastic radial basis functions and uncertainty quantification. *Struct. Multidiscip. Optim.* **2015**, *51*, 347–368. [[CrossRef](#)]
31. Iuliano, E. *Application of Surrogate-Based Global Optimization to Aerodynamic Design*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 25–46.

-
32. Fu, J.; Wang, L. A Random-Discretization Based Monte Carlo Sampling Method and its Applications. *Methodol. Comput. Appl. Probab.* **2002**, *4*, 5–25. [[CrossRef](#)]
 33. Safari, A.; Younis, A.; Wang, G.; Herpa, L.; Dong, Z. Development of a metamodel assisted sampling approach to aerodynamic shape optimization problems. *J. Mech. Sci. Technol.* **2015**, *29*, 2013–2024. [[CrossRef](#)]