

A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-Objective Optimization

Luis V. Santana-Quintero, Alfredo Arias Montaña
and Carlos A. Coello Coello *

CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07360, MEXICO
lvspenny@hotmail.com, aarias@computacion.cs.cinvestav.mx,
ccoello@cs.cinvestav.mx

Summary. Evolutionary algorithms have been very popular for solving multi-objective optimization problems, mainly because of their ease of use, and their wide applicability. However, multi-objective evolutionary algorithms (MOEAs) tend to consume an important number of objective function evaluations, in order to achieve a reasonably good approximation of the Pareto front. This is a major concern when attempting to use MOEAs for real-world applications, since we can normally afford only a fairly limited number of fitness function evaluations in such cases. Despite these concerns, relatively few efforts have been reported in the literature to reduce the computational cost of MOEAs. It has been until relatively recently, that researchers have developed techniques to achieve an effective reduction of fitness function evaluations by exploiting knowledge acquired during the search. In this chapter, we analyze different proposals currently available in the specialized literature to deal with expensive functions in evolutionary multi-objective optimization. Additionally, we review some real-world applications of these methods, which can be seen as case studies in which such techniques led to a substantial reduction in the computational cost of the MOEA adopted. Finally, we also indicate some of the potential paths for future research in this area.

1.1 Introduction

In many disciplines, optimization problems have, in a natural form, two or more objectives that we aim to minimize simultaneously, and which are normally in conflict with each other. These problems are called “multi-objective”, and their solution gives rise not to one, but to a set of solutions representing

* The third author is also affiliated to the UMI-LAFMIA 3175 CNRS.

the best possible trade-offs among the objectives (the so-called *Pareto optimal set*). In the absence of user's preferences, all the solutions contained in the Pareto optimal set are equally good. When plotted in objective function space, the contents of the Pareto optimal set produces the so-called *Pareto front*.

Evolutionary algorithms (EAs) have become a popular search engine for solving multi-objective optimization problems [17, 21], mainly because they are very easy to use and have a wide applicability. However, multi-objective evolutionary algorithms (MOEAs) normally require a significant number of objective function evaluations, in order to achieve a reasonably good approximation of the Pareto front, even when dealing with problems of low dimensionality. This is a major concern when attempting to use MOEAs for real-world applications, since in many of them, we can only afford a fairly limited number of fitness function evaluations.

Despite these concerns, relatively little efforts have been reported in the literature to reduce the computational cost of MOEAs, and several of them only focus on algorithmic complexity (see for example [36]), in which little else can be done because of the theoretical bounds related to nondominance checking [45].

It has been until relatively recently, that researchers have developed techniques to achieve a reduction of fitness function evaluations by exploiting knowledge acquired during the search [42]. Knowledge of past evaluations can also be used to build an empirical model that approximates the fitness function to optimize. This approximation can then be used to predict promising new solutions at a smaller evaluation cost than that of the original problem [40, 42]. Current functional approximation models include Polynomials (response surface methodologies [30, 65]), neural networks (e.g., multi-layer perceptrons (MLPs) [33, 34, 62]), radial-basis function (RBF) networks [60, 77, 83], support vector machines (SVMs) [4, 71], Gaussian processes [6, 78], and Kriging [24, 66] models. Other authors have adopted fitness inheritance [67] or cultural algorithms [46] for the same purposes.

In this chapter several possible schemes are described, in which the use of the knowledge from past solutions can help to guide the search of the new solutions, with particular emphasis on MOEAs. The remainder of this chapter is organized as follows. In Section 1.2, we present basic concepts related to multi-objective optimization. Then, in Section 1.3 we discuss several schemes that incorporate knowledge into the fitness evaluations of an evolutionary algorithm, providing a brief explanation of the surrogate models that have been used to approximate the fitness function. Next in Section 1.4 some selected research works are discussed. Such works are related to real-world engineering optimization problems, and can be considered as case studies in which the use of the described techniques led to a substantial reduction in the computational cost of the MOEA adopted. Finally, in Section 1.5, our conclusions and some potential paths for future research in this area are indicated.

1.2 Basic Concepts

The general multi-objective optimization problem (MOP) can be formally defined as the problem of finding:

$\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)^T$ which satisfies the m inequality constraints:

$$g_i(\vec{x}) \leq 0; \quad i = 1, \dots, m$$

the p equality constraints:

$$h_j(\vec{x}) = 0; \quad j = 1, \dots, p$$

and optimizes the vector function:

$$\mathbf{f}(\mathbf{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$$

In other words, we aim to determine from among the set S of all vectors (points) which satisfy the constraints those that yield the optimum values for all the k objective functions simultaneously. The constraints define the feasible region S and any point \vec{x} in the feasible region is called a feasible point.

1.2.1 Pareto dominance

Pareto dominance is formally defined as follows:

A vector $\vec{u} = (u_1, \dots, u_k)$ is said to dominate a vector $\vec{v} = (v_1, \dots, v_k)$ if and only if \vec{u} is partially less than \vec{v} , i.e., $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$ (assuming minimization).

In order to say that a solution dominates another one, it needs to be strictly better in at least one objective, and not worse in any of them. So, when we are comparing two different solutions A and B, there are 3 possible outcomes:

- A dominates B.
- A is dominated by B.
- A and B are incomparable.

1.2.2 Pareto optimality

The formal definition of *Pareto optimality* is provided next:

A solution $\vec{x}_u \in S$ (where S is the feasible region) is said to be *Pareto optimal* if and only if there is no $\vec{x}_v \in S$ for which $\mathbf{v} = f(\mathbf{x}_v) = (v_1, \dots, v_k)$ dominates $\mathbf{u} = f(\mathbf{x}_u) = (u_1, \dots, u_k)$, where k is the number of objectives.

In words, this definition says that \mathbf{x}_u is Pareto optimal if there exists no feasible vector \mathbf{x}_v which would decrease some objective without causing a simultaneous increase in at least one other objective (assuming minimization).

This definition does not provide us a single solution (in decision variable space), but a set of solutions which form the so-called *Pareto Optimal Set* (P^*). The vectors that correspond to the solutions included in the Pareto optimal set are *nondominated*.

1.2.3 Pareto front

When all nondominated solutions are plotted in objective function space, the nondominated vectors are collectively known as the *Pareto Front* (PF^*). Formally:

$$PF^* := \{ \vec{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_k(\mathbf{x})]^T \mid \mathbf{x} \in P^* \}$$

It is, in general, impossible to find an analytical expression that defines the Pareto front of a MOP, so the most common way to get the Pareto front is to compute a sufficient number of points in the feasible region, and then filter out the nondominated vectors from them.

The previous definitions are graphically depicted in Figure 1.1, showing the *Pareto front*, the *Pareto optimal set* and the *dominance* relations among solutions.

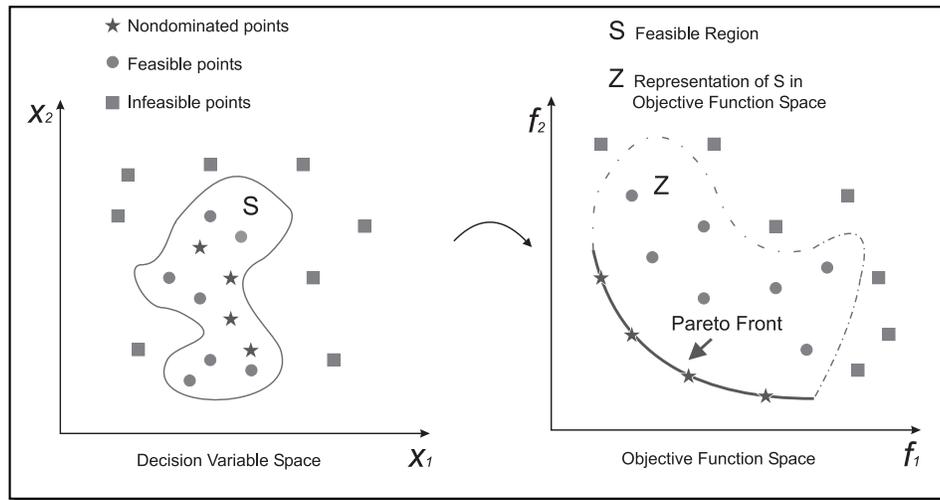


Fig. 1.1. Mapping of the Pareto optimal solutions to the objective function space

1.3 Knowledge Incorporation

From the many techniques adopted to solve such multi-objective optimization problems, evolutionary algorithms are among the most popular mainly

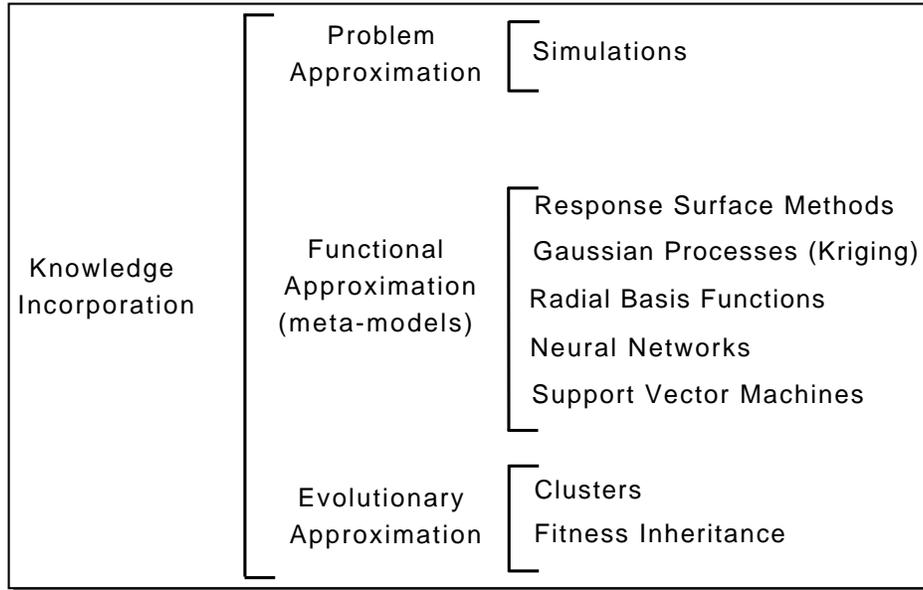


Fig. 1.2. A taxonomy of approaches for incorporating knowledge into evolutionary algorithms

because of their population-based nature, which is very useful to generate several nondominated solutions in a single run. However, dealing with a large population size and a large number of generations make MOEAs an unaffordable choice (computationally speaking) in certain applications, even when parallelism is adopted. In general, MOEAs can be unaffordable for an application when:

- The evaluation of the fitness functions is computationally expensive (i.e., it takes from minutes to hours).
- The fitness functions cannot be defined in an algebraic form (e.g., when the fitness functions are generated by a simulator).
- The total number of evaluations of the fitness functions is limited by financial constraints (i.e., there is a financial cost involved in computing the fitness functions).

Jin et al. [40] presented a taxonomy of approaches which incorporate knowledge into EAs (see Figure 1.2). From this taxonomy, we can distinguish three main types of strategies or approaches to deal with expensive fitness functions:

Problem approximation: Tries to replace the original statement of the problem by one which is approximately the same as the original problem but which is easier to solve. To save the cost of the experiments, numerical

simulations instead of physical experiments are used to pseudo-evaluate the performance of a design.

Functional approximation: In this case, a new expression is constructed for the objective function based on previous data obtained from the real objective functions. The models obtained from the available data are often known as *meta-models* or *surrogates* (see Section 1.3.1).

Evolutionary approximation: This approximation is specific for EAs and tries to save function evaluations by estimating an individual's fitness from other similar individuals. Two popular subclasses in this category are fitness inheritance and clustering.

1.3.1 Surrogates

In many practical engineering problems, we have black-box objective functions whose algebraic definitions are not known. In order to construct an approximation function, it is required to have a set of sample points that help us to build a meta-model of the problem. The objective of such meta-model is to reduce the total number of evaluations performed on the real objective functions, while maintaining a reasonably good quality of the results obtained. Thus, such meta-model is used to predict promising new solutions at a smaller evaluation cost than that of the original problem.

The accuracy of the surrogate model relies on the number of samples provided in the search space, as well as on the selection of the appropriate model to represent the objective functions. There exist a variety of techniques for constructing surrogate models (see for example [79]). One example is least-square regression using low-order polynomials, also known as response surface methods. Comparisons of several surrogate modeling techniques have been presented by Giunta and Watson [27] and by Jin et al. [39].

A surrogate model is built when the objective functions are to be estimated. This local model is built using a set of data points that lie on the local neighborhood of the design. Since surrogate models will probably be built thousands of times during the search, computational efficiency becomes a major issue of their construction process.

In [43], Knowles and Nakayama present a survey of meta-modeling approaches to solve specific problems. The authors discuss the problem on how to model each objective function and how to improve the Pareto approximation set using a trade-off method proposed by Nakayama et al. [56]. In multi-objective optimization problems, the trade-off method tries to satisfy an aspiration level at the k -th iteration, with the help of a trade-off operator which changes the k -th level if the decision maker (DM) is not satisfied with the solution. So, they combine the satisficing trade-off method and meta-modeling for supporting the DM to get a final solution with a low number of fitness function evaluations. They use the $\mu - v$ Support Vector Regression method [57] as their meta-model and include two real-world multi-objective optimization problems, using also a Radial Basis Function Network with a

Genetic Algorithm in searching the optimal value of the predicted objective function [58]. The proposed approach obtains good solutions within 1/10 or less analysis time than a conventional optimization approach based on a quasi-Newton method with approximated differentials.

1.3.2 Polynomials: response surface methods (RSM)

The response surface methodology comprises three main components: (1) regression surface fitting, in order to obtain approximate responses, (2) design of experiments in order to obtain minimum variances of the responses and (3) optimizations using the approximated responses.

An advantage of this technique is that the fitness of the approximated response surfaces can be evaluated using powerful statistical tools. Additionally, the minimum variances of the response surfaces can be obtained using design of experiments with a small number of experiments.

For most response surfaces, the functions adopted for the approximations are polynomials because of their simplicity, although other types of functions are, of course, possible. For the cases of quadratic polynomials, the response surface is described as follows:

$$\hat{y} = (\beta_0) + \sum_{i=1}^n (\beta_i \cdot x_i) + \sum_{i,j=1, i \leq j}^n (\beta_{i,j} \cdot x_i \cdot x_j) \quad (1.1)$$

where n is the number of variables, and β_0 and β_i are the coefficients to be calculated. To estimate the unknown coefficients of the polynomial model, both the least squares method (LSM) and the gradient method can be used, but either of them requires at least the same number of samples of the real objective function than the β_i coefficients in order to obtain good results.

1.3.3 Gaussian Process or Kriging

An alternative approach for constructing surrogate models is to use a Gaussian Process Model (also known as Kriging), which is also referred to as “Design and Analysis of Computer Experiments” (DACE) model [68] and Gaussian process regression [82]. This approach builds probability models through sample data and estimates the function values at every untested point with a Gaussian distribution.

In Kriging, the meta-model prediction is formed by adding up two different models as follows:

$$y(\vec{x}) = a(\vec{x}) + b(\vec{x})$$

where $a(\vec{x})$ represents the “average” long-term range behavior and the expected value of the true function. This function can be modeled in various ways, such as with polynomials or with trigonometric series as:

$$a(\vec{x}) = a_0 + \sum_{i=1}^L \sum_{j=1}^R a_{ij}(x_i)^j$$

where: R is the polynomial order with L dimensions and $b(\vec{x})$ stands for a local deviation term. $b(\vec{x})$ is a Gaussian random function with zero mean and non-zero covariance that represents a localized deviation from the global model. This function represents a short-distance influence of every data point over the global model. The general formulation for $b(\vec{x})$ is a weighted sum of N functions, $K_n(\mathbf{x})$ that represent the covariance functions between the n^{th} data point and any point \mathbf{x} :

$$b(\vec{x}) = \sum_{n=1}^N b_n K(h(x, x_n)) \text{ and } h(x, x_n) = \sqrt{\sum_{i=1}^L \left(\frac{x_i - x_{in}}{x_i^{max} - x_i^{min}} \right)^2}$$

where x_i^{min} and x_i^{max} are the lower and upper bounds of the search space and x_{in} denotes the i -th component of the data point x_n . However, the shape of $K(h)$ has a strong influence on the resulting aspect of the statistical model. That is the reason why it is said that Kriging is used as an estimator or an interpolator.

1.3.4 Radial basis functions

Radial Basis Functions (RBFs) were first introduced by R. Hardy in 1971 [32]. Let's suppose we have certain points (called centers) $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$. The linear combination of the function g centered at the points \vec{x} is given by:

$$f : \mathbb{R}^d \mapsto \mathbb{R} : \vec{x} \mapsto \sum_{i=1}^n \lambda_i g(\vec{x} - \vec{x}_i) = \sum_{i=1}^n \lambda_i \phi(\|\vec{x} - \vec{x}_i\|) \quad (1.2)$$

where $\|\vec{x} - \vec{x}_i\|$ is the Euclidean distance between the points \vec{x} and \vec{x}_i . So, f becomes a function which is in the finite dimensional space spanned by the basis functions:

$$g_i : \vec{x} \mapsto g(\|\vec{x} - \vec{x}_i\|)$$

Now, let's suppose that we already know the values of a certain function $H : \mathbb{R}^d \mapsto \mathbb{R}$ at a set of fixed locations $\vec{x}_1, \dots, \vec{x}_n$. These values are named $f_j = H(\vec{x}_j)$, so we try to use the \vec{x}_j as centers in the equation 1.2. If we want to force the function f to take the values f_j at the different points \vec{x}_j , then we have to put some conditions on the λ_i . This implies the following:

$$\forall j \in \{1, \dots, n\} f_j = f(\vec{x}_j) = \sum_{i=1}^n (\lambda_i \cdot \phi(\|\vec{x}_j - \vec{x}_i\|))$$

In these equations, only the λ_i are unknown, and the equations are linear in their unknowns. Therefore, we can write these equations in matrix form:

$$\begin{bmatrix} \phi(0) & \phi(\|x_1 - x_2\|) & \dots & \phi(\|x_1 - x_n\|) \\ \phi(\|x_2 - x_1\|) & \phi(0) & \dots & \phi(\|x_2 - x_n\|) \\ \vdots & \vdots & & \vdots \\ \phi(\|x_n - x_1\|) & \phi(\|x_n - x_2\|) & \dots & \phi(0) \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad (1.3)$$

Typical choices for the basis function $g(\mathbf{x})$ include linear splines, cubic splines, multiquadrics, thin-plate splines and Gaussian functions as shown in Table 1.1.

Type of Radial Function		
LS	linear splines	$ r $
CS	cubic splines	$ r ^3$
MQS	multiquadrics splines	$\sqrt{1 + (\epsilon r)^2}$
TPS	thin plate splines	$ r ^{2m+1} \ln r $
GA	Gaussian	$e^{-(\epsilon r)^2}$

Table 1.1. Radial basis functions

1.3.5 Artificial neural networks

An ANN basically builds a map between a set of inputs and the corresponding outputs, and are good to deal with nonlinear regression analysis with noisy signals [5]. A multilayer feedforward neural network consists of an array of input nodes connected to an array of output nodes through successive intermediate layers. Each connection between nodes has a weight, which initially has a random value, and that is adjusted during a training process. The output of each node of a specific layer is a function of the sum on the weighted signals coming from the previous layer. The crucial points in the construction of an ANN are the selection of inputs and outputs, the architecture of the ANN, that is, the number of layers and the number of nodes in each layer, and finally, the training algorithm.

The multi-layer perceptron (MLP) is a multilayered feedforward network that has been widely used in function approximation problems, because it has been often found to provide compact representations of mappings in real-world problems. An MLP is composed of neurons and the output (y) of each neuron is thus:

$$y = \phi \left(\sum_{i=1}^n w_i \cdot a_i + b \right)$$

where a_i are the inputs of the neuron, and w_i is the weight associated with the i^{th} input. The nonlinear function ϕ is called the activation function as it determines the activation level of the neuron.

In Figure 1.3, we show an MLP network with one layer of linear output neurons and one layer of nonlinear neurons between the input and output neurons. The middle layers are usually called *hidden layers*.

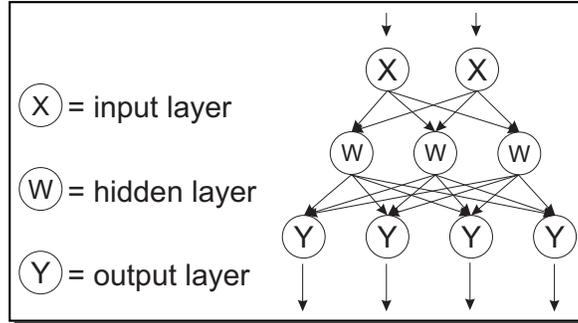


Fig. 1.3. A graphical representation of an MLP network with one hidden layer

To learn a mapping $\mathbb{R}^n \rightarrow \mathbb{R}^m$ by an MLP, its architecture should be the following: it should have n input nodes and m output nodes with a single or multiple hidden layer. The number of nodes in each hidden layer is generally a design decision.

Training an ANN

In general terms, supervised training consists of presenting to the network patterns whose output we know (the training set) finding the output of the net and adjusting the weights so as to make the actual output more like the desired (or teaching signal). The two most useful training protocols are: off-line and on-line. In off-line learning, all the data are stored and can be accessed repeatedly. In on-line learning, each case is discarded after it is processed and the weights are updated. With off-line learning, we can compute the objective function for any fixed set of weights, so we can see whether or not we are making progress in training.

Error backpropagation is the simplest and most widely used algorithm to train feedforward neural networks. In this algorithm the training is performed by minimizing a loss function, usually the sum of square errors over the N elements of the training set. In this case, it is adopted a generalization of the square error function given by:

$$J(W) = \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^c (t_{ki} - z_{ki})^2 = \frac{1}{2} \sum_{i=1}^N \|\vec{t}_i - \vec{z}_i\|^2$$

where \mathbf{t}_i and \mathbf{z}_i are the i^{th} -target and the i^{th} -network output vectors of length c , respectively; W represents all the weights in the network. The backpropagation learning rule is based on a gradient descent. The weights are initialized with random values, and are changed in a direction to reduce the error following the next rule:

$$W_{new} = W_{old} - \eta \frac{\partial J}{\partial W}$$

The weight update for the hidden-output weights is given by:

$$\partial W_{kj} = \eta(t_k - z_k)f'(net_k)y_j$$

and the input-to-hidden weights learning rule is:

$$\partial W_{ji} = \eta \cdot x_i \cdot f'(net_j) \sum_{k=1}^n w_{kj} \partial_k$$

where η is the learning rate, i, j, k are the corresponding node indexes for each layer and net_j is the inner product of the input layer with the weights w_{ji} at the hidden unit.

1.3.6 Support vector machines

Support vector machines (SVM) have become popular in recent years for solving problems in classification, regression and novelty detection. An important property of support vector machines is that the determination of the model parameters corresponds to a convex optimization problem, and thus, any local solution found is also a global optimum. In *SVM* regression, our goal is to find a function $f(x)$ that has at most an ϵ deviation from the obtained targets y_i for all the training data, and at the same time is as flat as possible. Let's suppose we are given training data $\chi = (x_t, y_t)_{t=1}^N$ where $y_t \in \mathbb{R}$. Then, the $f(x)$ is given by:

$$f(x) = \langle w, x \rangle + b \text{ with } w \in \mathbb{R}^d, x \in \mathbb{R}^d, b \in \mathbb{R}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product in χ . A small w means that the regression is flat. One way to ensure this, is to minimize the norm, $\|w\|^2 = \langle w, w \rangle$. The problem can be written as a convex optimization problem:

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|w\|^2 && (1.4) \\ & \text{subject to} && \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases} \end{aligned}$$

And one can introduce two slack variables ξ_i, ξ_i^* , for positive and negative deviations, $\xi_i \geq 0$ and $\xi_i^* \geq 0$, where $\xi_i > 0$ corresponds to a point for

which $\langle w, x_i \rangle + b > y_i + \epsilon$ and $\xi_i^* > 0$ corresponds to a point for which $\langle w, x_i \rangle + b < y_i - \epsilon$ (as in Figure 1.4):

$$\begin{aligned} & \text{minimize } C \sum_{i=1}^l (\xi_i + \xi_i^*) + \frac{1}{2} \|w\|^2 & (1.5) \\ & \text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned}$$

The constant $C > 0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated. The ϵ -insensitive loss function [80] (see equation (1.6)) means that we tolerate errors up to ϵ and also that errors beyond that value have a linear rather than a quadratic effect. This error function is therefore more tolerant to noise and is thus, more robust.

$$|\xi|_\epsilon = \begin{cases} 0, & \text{if } |\xi| \leq \epsilon; \\ |\xi| - \epsilon, & \text{otherwise.} \end{cases} \quad (1.6)$$

Figure 1.4, shows a plot of the ϵ -insensitive loss function. Note that only the points outside the shaded region contribute to the cost of the function. It turns out that in most cases, the optimization problem defined by equation (1.5) can be solved more easily in its dual formulation. The dual formulation also provides the capability for extending SVM to nonlinear functions using a standard dualization method based on Lagrange multipliers, as described by Fletcher [25]. So, optimizing the Lagrangian and substituting $t_i = \langle w, x_i \rangle$ for simplicity, we have:

$$\begin{aligned} L = C \sum_{i=1}^N (\xi_i + \xi_i^*) + \frac{1}{2} \|w\|^2 - \sum_{i=1}^N (\mu_i \xi_i + \mu_i^* \xi_i^*) \\ - \sum_{i=1}^N \alpha_i (\epsilon + \xi_i + y_n - t_n) - \sum_{i=1}^N \alpha_i^* (\epsilon + \xi_i^* + y_n - t_n) \end{aligned} \quad (1.7)$$

Then, we can substitute for $y(x)$ using the linear model equation: $y(x) = w^T \phi(x) + b$ and set the derivatives of the Lagrangian with respect to 1) w , 2) b , 3) ξ_i and 4) ξ_i^* to zero, giving:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \phi(x_i) \quad (1.8)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0 \quad (1.9)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i + \mu_i = C \quad (1.10)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i^* + \mu_i^* = C \quad (1.11)$$

Using these results to eliminate the corresponding variables from the Lagrangian, we see that the dual problem involves maximizing:

$$L'(a, a^*) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) - \epsilon \sum_{i=1}^N (\alpha_i + \alpha_i^*) + \sum_{i=1}^N (\alpha_i - \alpha_i^*) t_n \quad (1.12)$$

with respect to α_i and α_i^* , where $k(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j)$ is the kernel function. So, the problem becomes a constrained maximization problem with the box constraints:

$$\begin{aligned} 0 &\leq \alpha_i \leq C \\ 0 &\leq \alpha_i^* \leq C \end{aligned}$$

And the predictions for new inputs can be made using:

$$y(x) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) k(x, x_i) + b \quad (1.13)$$

The support vectors are those data points that contribute to predictions given by equation 1.13, in other words those for which either $\alpha_i \neq 0$ or $\alpha_i^* \neq 0$. These are points that lie on the boundary of the ϵ -tube or outside the tube. All points within the tube have $\alpha_i = \alpha_i^* = 0$.

1.3.7 Clustering

Clustering is the unsupervised classification of patterns into groups (or clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines [35].

Typical pattern clustering involves the following steps:

- (1) **Pattern representation:** it refers to the number of classes, number of patterns, and features available to the clustering algorithm.

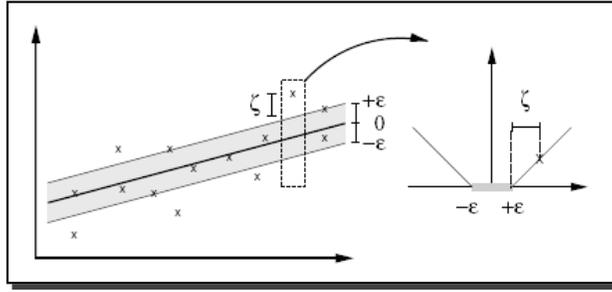


Fig. 1.4. ϵ -insensitive loss function for SVM

- (2) **Definition of a pattern proximity:** it is usually measured by a distance function defined on pairs. A simple distance measure such as the Euclidean distance can often be used to reflect dissimilarity between two patterns.
- (3) **Clustering or grouping:** it can usually be hard (a partition of the data into well-defined groups) or fuzzy (where each pattern belongs in certain degree to each of the output clusters).
- (4) **Data abstraction** (if necessary): it is the process of extracting a simple representation of a data set, and a compact description of each cluster, such as the centroid.
- (5) **Assessment of output** (if necessary): it distinguishes a good clustering result from a poor one, it attempts to study the cluster tendency, and it analyzes the clustering result with a specific criterion of optimality.

Although, there is no specific approach that uses only clustering to deal with the problem of reducing the number of objective function evaluations of a problem, clustering techniques are commonly used in combination with surrogates. The computational cost of a surrogate method can become prohibitively high when the size of the training data set is very large, because of the time that it could require to process the data set. In such cases, it is common to cluster the whole data set into several small clusters and then try to build an independent local model from them.

1.3.8 Fitness Inheritance

Fitness Inheritance is a technique that was introduced by Smith et al. [72], whose main motivation is to reduce the total number of fitness function evaluations performed by an evolutionary algorithm. The mechanism works as follows: when assigning the fitness to an individual, some times we evaluate the objective function as usual, but the rest of the time, we assign fitness as an average of the fitness of the parents. This saves one fitness function evaluation, and is based on the assumption of similarity of an offspring to its parents.

Fitness inheritance must not be always applied, since the algorithm needs to use the true fitness function several times, in order to obtain enough information to guide the search. The percentage of time in which fitness inheritance is applied is called *inheritance proportion*. If this inheritance proportion is 1, the algorithm is most likely to prematurely converge [8].

It is important to mention that some researchers consider this mechanism not so useful in complex or real world problems, under the argument that it has been only applied in “easy” problems. For example, Ducheyne et al. [23] tested the original scheme of fitness inheritance on a standard binary genetic algorithm and the Zitzler-Deb-Thiele (ZDT) [84] multiobjective test problems, concluding that fitness inheritance was not useful when dealing with difficult shapes of the Pareto front. Other authors, however, have successfully applied fitness inheritance to the ZDT and other (more complicated) test problems (see for example [67]).

1.4 Real-World Applications

In this section, we present some selected research work in which a real-world multi-objective engineering optimization problem was solved using a MOEA coupled to a technique for reducing the computational cost involved. There are many engineering disciplines which require expensive function evaluations. From them, we chose aeronautical/aerospace engineering, because it presents problems having high CPU time demand, high nonlinearity and, some times, also high dimensionality. All of these features are also common in other engineering optimization problems, and we consider them representative of the main sources of difficulty in engineering optimization in general.

Aeronautical and aerospace engineering are disciplines in which the solution of multi-objective/multi-disciplinary problems is a standard practice. During the last three decades, the process of engineering design in these industries has been revolutionized as computational simulation has come to play an increasingly dominant role. The increasing demand of optimal and robust designs, driven by time to market, economics and environmental constraints, along with the increasing computing power available, has changed the role of computational simulations from being used only as analysis tools to be used as design optimization tools.

Among the problems with expensive evaluations identified in these disciplines are the following:

- **Aerodynamic Shape Optimization:** This type of optimization problem ranges from $2D$ to complex $3D$ shapes. Typical optimization applications for $2D$ problems comprise Wing and Turbine Airfoil Shape Optimization as well as Inlet/Nozzle design optimization, whereas for $3D$ problems, turbine blade, Wing Shape and Wing-Body configuration design optimizations are typical example applications.

- **Structural Optimization:** The aeronautical/aerospace design philosophy focuses on the design of structures with minimum weight that are strong enough to withstand certain design loads. These two objectives are conflicting in nature and, therefore, the aim of structural optimization is to find the best possible compromise between them. Typical applications for this type of problems comprise structural shape and topology optimization, robust structural design and structural weight optimization.
- **Multidisciplinary Design Optimization:** aeronautical/aerospace design has a multidisciplinary nature, since in many practical design applications, two or more disciplines are involved, each one with specific performances to accomplish. Typical applications for this type of problems are the aeroelastic applications in which aerodynamics and structural engineering are the interacting disciplines.

For all the optimization problems indicated above, the objective function evaluations are routinely done by using complex computational simulations such as CFD (Computational Fluid Dynamics) in the case of aerodynamic problems, CAA (Computational Aero-Acoustics) for aero-acoustic problems, CSM (Computational Structural Mechanics, by means of Finite Element Method software) for Structural Optimization Problems, or a combination of them in the case of multidisciplinary design optimization problems. Because of their nature, any of these computational simulations have a high computational cost (since they solve, in an iterative manner, the set of partial differential equation governing the physics of the problem) and evaluating the objective functions for the kind of problems indicated above, can take from minutes to hours for a single candidate solution, depending on the fidelity of the simulation.

Nowadays in aeronautical/aerospace industries, MOEAs have gained popularity and are considered as a mature and reliable numerical optimization tool, since they provide to the designers not only with one design solution, but with a set of them from which the tradeoff between the competing objectives can be assessed. This last situation can help decision makers to select a compromise design according to his/her own preferences. Given the high computational cost required for the computational simulations and the population based nature of MOEAs, the use of hybrid methods or meta-models is a natural choice in order to reduce the computational cost of the design optimization process, as indicated by some representative research works that will be described next.

1.4.1 Use of problem approximation

As indicated in Section 1.3, this approach tries to replace the original problem by one which is approximately the same as the original one but which is easier to solve. In the context of aeronautical/aerospace engineering problems, where complex CFD, CAA and CSD are employed, the problem can be approximated by using different resolutions in the flow or structural simulation

by using either coarse or fine grids. In the case of CFD simulations another level of approximation can be obtained by solving Euler flows or potential flows instead of Navier-Stokes flow simulations. Some of these techniques are used in the following research works.

Chiba et al. [9, 10] addressed the problem of multidisciplinary wing shape optimization using the ARMOGA (Adaptive Range Multi-Objective Genetic Algorithm) [69] and CFD and CSD Simulations. Three objective functions are minimized: (i) Block Fuel, (ii) Maximum takeoff weight, and (iii) Difference in the drag coefficient between transonic and subsonic flight conditions. In this work, and during the optimization process, an iterative aeroelastic solution is performed in order to minimize the wing weight, with constraints on flutter and strength requirements. For this iterative process, Euler flow solutions (instead of Navier-Stokes flow solutions) are used as a problem approximation in order to reduce the computational cost. Also, a flight envelope analysis is done, which uses high-fidelity CFD Navier-Stokes flow solutions for various flight conditions. The whole optimization process evolves a population of 8 individuals during 16 generations. Authors indicate that they use on the order of 70 Euler and 90 Navier-Stokes simulations per generation of their MOEA.

Sasaki et al. [69, 70] and Obayashi and Sasaki [59], solved a supersonic wing shape optimization problem minimizing four objective functions: (i) drag coefficient at transonic cruise, (ii) drag coefficient at supersonic cruise, (iii) bending moment at the wing root at supersonic cruise condition, and (iv) pitching moment at supersonic cruise condition. In this research study, which also makes use of the ARMOGA algorithm, no iterative aeroelastic analysis is performed, aiming at reducing the associated computational cost. The objective associated with the bending moment at wing root, is approximated by numerical integration of the pressure distribution over the wing surface, as obtained by the CFD analysis.

Lee et al. [48, 51] presented the application of the HAPMOEA (Hierarchical Asynchronous Parallel Multi-Objective Evolutionary Algorithm) [31] to the robust design optimization of an ONERA M6 wing shape. The optimization problem is solved considering uncertainties in the design environment, related to the flow Mach number. The Taguchi method is employed to transform the problem into one with two objectives: (i) minimization of the mean value of an objective function with respect to variability of the operating conditions, and (ii) minimization of the variance of the objective function of each solution candidate, with respect to its mean value. HAPMOEA is based on evolution strategies, incorporating the concept of the Covariance Matrix Adaptation (CMA). It also incorporates a Distance Dependent Mutation (DDM) operator, and a hierarchical set of CFD models (varying the grid resolution of the solver) and populations; small populations are evolved using fine mesh CFD solutions (exploitation of solutions) while large populations are evolved with coarse mesh CFD solutions (exploration of solutions). Good solutions from the coarse mesh populations are transferred to the fine mesh populations. The use of a hierarchical set of CFD models can be seen as dif-

ferent levels of fitness approximation; low-quality fitness approximations are obtained by using coarse mesh grids at low computational cost, while high-quality fitness approximations are obtained by using a fine mesh grid with its associated higher computational cost.

Lee et al. [49, 50] made use of a generic framework for multidisciplinary design and optimization [31] to explore the application of a robust MOEA-based algorithm for improving the aerodynamic and radar cross section characteristics of an UCAV (Unmanned Combat Aerial Vehicle). In both applications, two disciplines are considered, the first concerning the aerodynamic efficiency and the second one dealing with the visual and radar signature of an UCAV airplane. The evolutionary Algorithm employed corresponds to the HAPMOEA indicated above. In this case, the minimization of three objective functions is considered: (i) inverse of the lift/drag ratio at ingress condition, (ii) inverse of the lift/drag ratio at cruise condition, and (iii) frontal area. The problem has, approximately, 100 decision variables, and the first two objective functions are evaluated using a potential flow solver (FLO22) coupled to FRICTION code for obtaining the viscous drag. The use of these last two codes approximates the Navier-Stokes flow solution, considerably reducing the computational cost. The evolutionary system evaluates a total of 1600 solution candidates from which, a Pareto set containing 30 members is obtained. From these nondominated solutions, a single compromise solution is obtained. The authors reported a solution time of 200 hours on a single processor.

1.4.2 Use of RSM by polynomial approximation

Lian and Liou [52] used a multi-objective genetic algorithm coupled to a second-order polynomial response surface model for the multiobjective optimization of a three-dimensional rotor blade. The optimization problem consisted of the redesign of the NASA rotor 67 compressor blade, a transonic axial-flow fan rotor which acts as the first stage of a two-stage compressor fan. Two objectives are considered: (i) maximization of the stage pressure raise, and (ii) minimization of the entropy generation. A constraint is imposed on the mass flow rate to have a difference less than 0.1% between the new and the reference design. Blade geometry is constructed from airfoil shapes defined at four span stations, with 32 total design variables. The quadratic response surface model is constructed with 1,024 sampling design candidates and using the IHS (Improved Hypercube Sampling) algorithm [3]. The authors noted that the evaluation of the 1,024 sampling individuals took approximately 128 hours (5.3 days) using eight processors and a Reynolds-Averaged Navier-Stokes CFD simulation. The optimization process for this application is done for 200 generations with a population size of 320 individuals, where objective functions are obtained from the approximation model. Following the optimization process, 12 design solutions are selected from the obtained response surface method Pareto front, and verified with the high fidelity CFD simulation. Objective functions differ slightly from those obtained

using the approximation model, and all selected solutions are better in both objective functions than the reference design. A similar research work is presented by Lian and Liou [53, 54], but minimizing the blade weight instead of entropy generation.

Goel et al. [29] used a quintic polynomial response surface method for solving a liquid-rocket injector multiobjective optimization design problem. Four competing objectives are considered: i) combustion length, ii) injector face temperature, iii) injector wall temperature, and iv) injector tip temperature. In this research, the NSGA-IIa (referred to as archiving NSGA-II [22]), and a local search strategy called “ ϵ – *constraint*” are adopted to generate a solution set that is used for approximating the Pareto optimal front by a response surface method (RSM). Once the Pareto optimal solutions are obtained, a clustering technique is used to select representative tradeoff design solutions.

Pagano et al. [61] presented an application for three-dimensional aerodynamic shape optimization, particularly the aerodynamic shape of an aircraft propeller. The aim of this multiobjective optimization is to improve an actual propeller performance. The authors considered two conflicting objectives: (i) minimize noise emission level, and (ii) maximize aerodynamic propeller efficiency. For this industrial problem, several disciplines are considered and, therefore the objective function evaluations consider: (a) aerodynamics, (b) structural behavior, and (c) aeroacoustics. For each of these, specialized computer simulation codes are employed. Every calculation comprises an iterative coupling procedure (fluids-structures-acoustics) among these simulation codes in order to evaluate a more realistic operating condition. As a consequence, the optimization process becomes computationally demanding. In order to reduce the burden of this high computational cost, the authors made use of design of experiment techniques (DOE), and a quadratic response surface method (RSM) for efficiently exploring the design space. The geometry for the propeller blade is parameterized using a total of 14 design variables. The optimization problem contains constraints on the geometry design variables and on propeller shaft power at two flight conditions; takeoff and cruise, respectively. The evolutionary algorithm employed corresponds to the NSEA+ (Nondominated Sorting Evolutionary Algorithm) as implemented in the OPTIMUS commercial code which is adopted by the authors. The population size for the evolutionary algorithm is set to 20 individuals, and the optimization is run using the DOE and RSM methods. Afterwards, the Pareto front solutions obtained are evaluated using the high fidelity simulation codes. The authors indicated that a total of 340 designs were evaluated with using high fidelity simulations. From them, approximately 20 Pareto solutions were obtained, all of them being better than the reference design in the two objectives considered.

1.4.3 Use of Artificial Neural Networks

Rai [64] addressed the problem of multiobjective robust design of a turbine blade airfoil, considering performance degradation due to manufacturing uncertainties. For this problem, the objectives are: (i) minimize the variance of the pressure distribution over the airfoil's surface, and (ii) maximize the wedge angle at the trailing edge. Both objectives must be met subject to the constraint that the required flow turning angle is achieved. Objectives are evaluated by means of a model that modifies the geometry of the airfoil surface following a probability density function that is observed for manufacturing tolerances, and with a CFD simulation for obtaining the flow pressure distribution. The blade geometry is defined by eight design parameters, but only two of them are varied during the optimization process. The evolutionary algorithm used in this research correspond to a multiobjective version of the differential evolution algorithm previously implemented by the same author and described in [63]. In order to cope with the associated calculation time of the CFD simulations required to evaluate the objective functions, the authors used a *hybrid neural network* comprised of 10 individual *single-hidden-layer feed forward networks*. The optimization is run with a small population size of 10 individuals and during 25 generations.

Arabnia and Ghaly [2] presented a strategy that makes use of multi-objective evolutionary algorithms for aerodynamic shape optimization of turbine stages in three-dimensional fluid flow. The NSGA [74] is used and coupled to an artificial neural network (ANN) based response surface method (RSM) in order to reduce the overall computational cost. The blade geometry, both for rotor and stator blades, is based on the E/TU-3 turbine which is used as a reference design to compare the optimization results to. The multi-objective optimization consists of finding the best distribution of 2D blade sections in the radial and circumferential directions. For this, a quadratic rational Bèzier curve, with 5 control points, is used for each of the two blades. The objective functions to be optimized include: (i) maximization of isentropic efficiency for the stage, and (ii) minimization of the streamwise vorticity. Both objective functions are evaluated using a 3D CFD flow simulation with constraints on: (1) inlet total pressure and temperature, (2) exit pressure, (3) axial chord and spacing, (4) inlet and exit flow angles, and (5) mass flow rate. The authors noted that one CFD simulation took approximately 10 hours. Therefore they resorted to an ANN based RSM. The ANN model with backpropagation, containing a single hidden layer with 50 nodes, was trained and tested with 23 CFD simulations, sampling the design space using the latin hypercubes sampling technique. The optimization process used the ANN model to estimate the objective functions, and the constraints values as well. The population size used in the NSGA was set to 50 individuals, and was run for 150 generations. Finally, the Pareto solutions were evaluated with the CFD flow simulation. From their results, the authors indicated that they obtained design solutions which were better in comparison to the reference turbine design. Indeed, they

attained a 1.2% improvement in stage efficiency, which is remarkable considering the small number of design variables used in the optimization process.

Alonso et al. [1] described a procedure for the multi-objective optimization design of a generic supersonic aircraft. The competing design objectives considered were two: i) maximization of aircraft range, and ii) minimization of the perceived loudness of the ground boom signature. Constraints were set for aircraft's structural integrity, take-off field length and landing field length. The objective functions were evaluated using CFD with various fidelity (approximation) levels. In this work, the authors made use of a neural network (NN) based response surface method. The prototype for the NN is a single hidden layer perceptron with sigmoid activation functions, providing a general nonlinear model, which is useful for the high non-linearities present in the objective functions landscapes associated to this problem. The neural network was trained with 300 sampling design solutions, obtained with low fidelity simulations in order to reduce the computational cost. In their optimization cycle, authors used high fidelity simulations only in promising regions of the design space to do a local exploration. The problem comprised 10 design variables and the NSGA-II [22] was used as the search engine with a population size of 64 and was run for 1000 generations using the surrogate-based objective function.

1.4.4 Use of a Gaussian Process or Kriging

D'Angelo and Minisci [20] used an evolutionary algorithm based on MOPED [19], which is a multi-objective optimization algorithm for continuous problems that uses the Parzen method to build a probabilistic representation of Pareto solutions, with multivariate dependencies among variables. The authors included three modifications to improve a previous implementation of MOPED: (a) use of a kriging model by which solutions are evaluated without resorting to costly computational simulations, (b) use of evolution control, which is adopted to avoid the evolution to converge to a false minima; the mechanism of this technique is to evaluate a subset of individuals or the whole actual generation, with the real simulation model, for a continuous kriging model update, and (c) hybridization of the algorithm; in this case, the selection and ranking of the individuals is different from the original algorithm and some mechanisms borrowed from the NSGA-II algorithm are adopted as well. In their optimization examples, subsonic airfoil shape optimization was performed. The optimization problem considered two objective functions: (i) drag force coefficient, and (ii) lift force coefficient difference with respect to a reference value. Both objectives were minimized. The airfoil geometry is parameterized using Bèzier curves both for its camber line and thickness distribution. In total, 5 design variables were used and constraints were imposed on the objective functions extreme values. The authors indicated that the subsonic airfoil shape optimization presented several difficulties. For example, the true Pareto front was discontinuous and partially converged solutions from the

aerodynamic simulation code introduced irregularities in the objective function. It is important to note that the approximation model used (*kriging*) reduced the number of real evaluations to only 2300, considering that the evolution system comprised a population size of 100 individuals and a total of 150 generations.

Song and Keane [73] applied a multi-objective genetic algorithm for studying the shape optimization of a civil aircraft engine nacelle. The primary goal of the study was to identify the tradeoff between aerodynamic performance and noise effects associated with various geometric features for the nacelle. The geometry was parameterized using 40 parameters, 33 of which were considered as design variables. In their study, the authors used NSGA-II [22] as the multi-objective search engine, while a commercial software was used for the CFD evaluations of the three-dimensional flow. Due to the large size of the design space to be explored, as well as the simulations being very time consuming, a kriging surrogate model was adopted in order to keep to a minimum the number of designs being evaluated with the CFD tool. The kriging model was continuously updated, adding sampling solutions from the Pareto front obtained using the kriging model and evaluated with the CFD tool. In their research, the authors reported difficulties in obtaining a converged Pareto front (there exist large discrepancies between the approximated and the real Pareto fronts). They attributed this behavior to the large number of variables in the design problem, and to the associated difficulties in obtaining an accurate kriging model for these situations. In order to alleviate this situation, they performed an ANOVA (Analysis of Variance) test to find the variables that contributed the most to the objective function values. After this test, they presented results with a reduced kriging surrogate model, employing only 7 variables. The authors argued that they obtained a similar design with this reduced kriging model at a considerably lower computational effort.

Jeong et al. [38] investigated the improvement of the lateral dynamic characteristics of a lifting-body type re-entry vehicle in transonic flight condition. The problem was posed as a multi-objective optimization problem in which two objectives were minimized: (i) derivative of the yawing moment, and (ii) derivative of the rolling moment. Due to the geometry of the lifting body and the operating flow condition of interest, namely high Mach number and strong vortex formation, the evaluation of the objectives was done by means of a full Navier-Stokes CFD simulation. Since the objectives were derivatives, multiple flow solutions were required to determine their values in a discrete manner through the use of finite differencing techniques. This considerably increased the total computational time due to a large number of calls for the CFD code. The optimization problem considered 4 design variables, and two solutions were sought: the first one without constraints, and the second one constraining the L/D ratio for the lifting-body type reentry vehicle. The authors used the EGOMOP (Efficient Global Optimization for Multi-Objective Problems) algorithm developed by Jeong et al. [37]. Such algorithm was built upon the ideas borrowed from the EGO and ParEGO algorithms from Jone et al. [41]

and Knowles et al. [42], respectively. EGOMOP adopts the use of the kriging model as a response surface model, for predicting the function value and its uncertainty. For the exploration of the Pareto solutions, Fonseca's MOGA [26] was used. The initial kriging model was built by using the latin hypercube sampling method for uniformly covering the design space, and the model was continuously updated.

Voutchkov et al. [81] used the NSGA-II [22] to perform a robust structural design of a simplified FEM jet engine model. This application aimed at finding the best jet engine structural configuration minimizing: the variation of reacting forces under a range of external loads, the mass for the engine and the engine's fuel consumption. These objectives are competing with each other and, therefore, the authors used a multi-objective optimization technique to explore the design space looking for trade-offs among them. The evaluation of the structural response was done in parallel by means of finite element simulations. The FEM model comprised a set of 22 groups of shell elements. The thickness for 15 of these groups were considered as the design variables. Computational time was reduced by using a kriging based response surface method. The optimization problem was posed as a MOP, comprising four objectives (all to be minimized): (i) standard deviation of the internal reaction forces, (ii) mean value of the internal reaction forces, (iii) engine's mass, and (iv) mean value of the specific fuel consumption. The first two objectives were computed over 200 external load variations. The authors noted that for this class of problem which comprises huge combinations of loads and finite element thicknesses, the multiobjective optimization problem would take on the order of one year of computational time on a single 1 GHZ CPU. Also, they indicated that by using the surrogate model and parallel processing, the optimization time was reduced to about 26 hours in a cluster with 30 PEs (processing elements).

Todoroki and Sekishiro [75, 76] proposed a new optimization method for composite structural components. This approach is based on the use of a multi-objective genetic algorithm coupled to a kriging model, in order to reduce the number of objective function evaluations, and to a FBB (Fractal Branch and Bound) method for the stacking sequence optimization needed in laminar composite structures. The problem consisted of two objectives: (i) minimize the structural weight of a hat-stiffened wing panel, subject to buckling load constraints, and (ii) maximize the probability of satisfying a predefined buckling load. The variables for the problem are a set of mixed real/discrete variables. Real variables correspond to the stiffener geometry definition, while discrete variables correspond to the number of plies for the composite panel. Constraints were imposed on the dimensions of the stiffener, but they were automatically satisfied in the definition of the variables ranges. The authors noted that the buckling load constraint demanded a large computational cost, since it needed a FEM (Finite Element Analysis). For this reason a kriging model was adopted and initialized with sampling points obtained by the LHS (Latin Hypercube Sampling) technique. The optimization

cycle consisted of two layers. The upper one driven by the multi-objective genetic algorithm and the kriging model, in which the optimization of the structural dimensions was performed. In the lower layer, the stacking sequences of the stiffener and panels were optimized by means of the FBB method. The evolutionary algorithm was run for 300 generations with a population of 100 individuals, and every 50 generations some nondominated solutions were evaluated with the FEM model, in order to update the kriging model. The authors obtained a Pareto Front that was discontinuous. Also, from the results obtained, a comparison of different designs was made. The solution obtained with the evolutionary algorithm was 3% heavier than a previous design obtained with a conventional method (deterministic), but obtained after only 301 FEM analyses compared to the tens of thousands required by the conventional method.

Choi et al. [11] used the NSGA-II [22] in the solution of a multidisciplinary supersonic business jet design. In this case, the disciplines involved were (i) aerodynamics and, (ii) aeroacoustics. The main objective of this particular problem was to obtain a compromise design having good aerodynamic performance while minimizing the intensity of the sonic boom signature at the ground level. Multiobjective optimization was used to obtain tradeoffs among the following objectives: (i) the aircraft drag coefficient, (ii) initial pressure raise (boom overpressure), and (iii) ground perceived noise level. All the objectives were minimized. The geometry of the aircraft was defined by 17 design variables, involving the modification of the wing platform, its position along the fuselage, and some cross sections and camber for the fuselage. For evaluating the objective functions, a high fidelity Euler solution was obtained with a very fine grid close to the aircraft's surface. In order to reduce the computational time required for the optimization cycle, a kriging model was employed. Its initial definition was formed with a latin hypercube sampling of the design space with 232 initial solutions, including both feasible and infeasible candidates. Following a kriging based optimization cycle, the Pareto optimal solutions were evaluated with high fidelity simulation tools and used to update the kriging model. In the example, constraints were imposed on some geometry parameters, and on the aircraft's operational conditions. No special constraint-handling mechanism was adopted other than discarding the solution candidates that did not satisfy the constraints, which were mostly geometrical. From their results, the authors noted that after the first design cycle using the kriging based NSGA-II, 59 feasible solutions were obtained. It is important to note that all the solutions obtained were better in both objectives compared to a base design. Another important issue in this particular application was that the kriging model did not perform as well as in other applications. The reason for this behavior was the high nonlinear physics involved in the two disciplines considered, which required, in consequence, more design cycles in the optimization.

In related work, Chung and Alonso [12] and Chung et al. [13] solved the same previously defined multidisciplinary problem, but using the μ -GA Algo-

rithm, from Coello Coello and Toscano Pulido [15, 16]. This change was aimed at reducing the total number of function evaluations during the optimization process. This μ -GA algorithm used a population size of 3 to 6 individuals and an external file to keep track of the nondominated solutions obtained so far. In the study reported in [12], the design cycles were performed using a kriging model. Two design cycles were executed, each one consisting of 150 solution candidates using the latin hypercube sampling technique applied around a base design in the first cycle. For the second cycle, the sampling was applied around the best solution obtained in the previous cycle. The authors reported that they obtained a very promising Pareto front estimation with only 300 functions evaluations. In the second study, reported in [13], the authors proposed and tested the GEMOGA (Gradient Enhanced Multiobjective Genetic Algorithm). The basic idea of this algorithm is to enhance the Pareto solutions with a gradient based search. One important feature of the algorithm is that gradient information is obtained from the kriging model. With this, the computational cost is not considerably increased.

Kumano et al. [44] used Fonseca's MOGA [26] for the multidisciplinary design optimization of wing shape for a small jet aircraft. In this study, four objectives were considered: (i) drag at the cruise condition, (ii) drag divergence between cruising and off-design condition, (iii) pitching moment at the cruising condition, and (iv) structural weight of the main wing. All these objectives were minimized. In this study, the optimization process was also performed by means of a kriging model, and such model was continuously updated after a certain prescribed number of iterations (defined by the user), adding new nondominated points obtained from the optimization steps.

1.4.5 Use of Clustering

Langer et al. [47] applied an integrated approach using CAD (Computer Aided Design) modeling with a MOEA for structural shape and topology optimization problems. The application presented in this research, dealt with the structural optimization of a typical instrument panel of a satellite, and considered two objectives: (i) minimize the instrument panel mass, and (ii) maximize the first eigenfrequency. The problem contained a mixed continuous/discrete set of variables. 17 design variables were used, from which 3 were discrete, which consider the number of stringers to use in the panel, as well as the plate and stringer materials. The authors solved the optimization problem for three shape and topology optimization cases: (a) a panel without instruments, (b) a panel with instruments at fixed positions, and (c) a panel with instrumental placing. They made use of polynomial based response surface methods in order to reduce the computational cost. Multiple local approximation models were constructed using a clustering technique. In all the examples included, the population size was set to 200 and was evolved for 20 generations. The evaluation of the objective functions comprised four load cases: (a) quasi-static acceleration, (b) modal analysis, (c) sinusoidal vibration loads, and (d)

‘pseudo temperature’ load. This latter load case, constrained the positioning of the instruments on the panel, since it imposed a limiting operating temperature for a specific instrument. The first three load cases were evaluated in parallel using FEM (Finite Element Method) simulations on a cluster of 32 workstations.

1.4.6 Use of Radial Basis Functions

Cinnella et al. [14] presented the airfoil shape optimization for transonic flows of BZT (Bethe-Zel’dovich-Thompson) fluids, by using a multi-objective genetic algorithm. This application explored the design of airfoil shapes in turbine cascades which could exploit the benefits of BZT transonic flows past airfoils. In the application, the authors proposed two optimization problems which aimed at finding optimal airfoil geometries both for (i) non-lifting airfoils, and (ii) lifting airfoils. In both cases GA-Based approaches were used as search engines. In the second case, the optimization problem considered two design objectives: (i) maximize lift at BZT subcritical conditions, and (ii) Minimize wave drag while maximizing lift for supercritical BZT flow conditions. Therefore, a bi-objective problem was solved, and the evolutionary algorithm helped the designers to find trade-off solutions between these two design points. The multi-objective genetic algorithm used in the second case was the NSGA [74]. In previous related work [18], a population size of 36 and 24 generations were used (totaling 864 objective function evaluations obtained from CFD), based on the constraint that the whole CFD calculation time had to be kept inferior to one week (the evaluation time for each individual varied from 5 to 10 min in a PC equipped with a Pentium Processor). In order to reduce the computational cost, the authors included an ANN (Artificial Neural Network) based on radial basis functions, formed by an input layer, an intermediate layer, and an output layer. The weights of the linear combinations were determined through a training procedure. The number of neurons involved was taken as the number of individuals in the training set. The first training set was formed with all the solutions obtained from the first two generations. Afterwards, the objective functions were approximated with the ANN-RBF model, and the training set was updated by adding a 30% of “exactly evaluated” individuals per generation. With this technique the authors obtained similar design solutions with approximately 60% less computational cost.

Kampolis and Giannakoglou [7] solved the inverse design of an isolated airfoil at two operating conditions. For this design problem, two reference airfoil and operating conditions were defined (these solutions could be seen as the extreme portions of the Pareto front), and a MOEA was used to find the tradeoff solutions between them. The MOEA adopted was SPEA-2 [85]. In their approach, the authors proposed the use of a radial basis function meta-model.

1.5 Conclusions and Future Research Paths

We have described several techniques which have been coupled to MOEAs, aiming to reduce the computational cost of evaluating the objective functions of a multi-objective optimization problem. Additionally, some selected real-world applications of such techniques were also presented as case studies in which these hybrid schemes led to substantial reductions in the computational cost. The main aim of this review was to provide a general overview of this area, which we believe that may be of interest both for MOEA researchers who may be looking for new algorithmic design challenges, and for practitioners, who may benefit from combining MOEAs with surrogate methods or any other approximation techniques that they normally use to reduce the computational cost of their simulations.

From the application examples reviewed here, we observed that the most preferred methods seem to be problem approximation, kriging and polynomial interpolation, followed by the use of neural networks and radial basis functions. Our study of the small sample of real-world applications presented here, also led us to outline some of the future research paths that seem promising within this area:

- **Model selection guidelines:** Since the high computational cost involved in applications such as those described here preclude us from any exhaustive experimentation, the existence of guidelines that allow us to identify which sort of method could be a good choice for a given problem would be of great help. To the authors' best knowledge no guidelines of this sort have even been reported in the specialized literature.
- **Hybridization:** Approximation models can be used not only to replace the objective function evaluations, but also to estimate first-order information (e.g., gradient information). This could lead to the use of hybrids of MOEAs with gradient-based methods. An example of this type of approach is presented in Chung et al. [13], where solutions are improved by the use of gradient information obtained from a kriging model. This sort of hybridization scheme is, however, relatively scarce in the literature until now.
- **Use of multiple approximation models:** Most authors report the use of a single approximation model. However, it may be worth exploring the combination of several of them for exploiting either their global or their local nature. This idea has been explored in the past, for example, by Mack et al. [55], by using a combination of polynomial response surface methods and radial basis functions, for performing global sensitivity analysis and shape optimization of bluff bodies. Also, Glaz et al. [28] adopted three approximation models, namely polynomial, kriging, and radial basis functions. This combined approach, adopted a weighted estimation from the different models, which was used to reduce the vibration for a helicopter rotor blade. To the authors' best knowledge, no similar combination of approaches has ever been reported when using MOEAs.

- **Automatic Switching:** Considering that every approximation model has particular properties in terms of global or local accuracy, and that the selection of the “best” approximation method to use for a particular application can also be considered a difficult task, one promising research area is to develop mechanisms allowing to automatically switch from one approximation method to a different one, as the optimization process is being executed. For example, a global approximation method (i.e., coarse-grained) could be used for exploration of the design space, while a more locally accurate method (i.e., fine-grained) might be used for solution exploitation.
- **Sampling techniques:** The accuracy of the approximation highly depends on the sampling and updating technique used. In most cases, the initial sampling is defined by a latin hypercube sampling, aiming at covering as much as possible the design space. This can be considered as a general technique. Another possibility is to use application-dependent sampling techniques, where the initial sampling design points are selected on the basis of reference or similar solutions. One example of this sort of situation is reported by Chung et al. [13] and by Chung and Alonso [12], where the initial approximation models are built around a reference design in decision variable space.

References

1. Alonso J, LeGresley P, Pereyra V (2008) Aircraft Design Optimization. *Mathematics and Computer in Simulation* 79:1948–1958
2. Arabnia M, Ghaly W (2008) A strategy for multi-objective shape optimization of turbine stages in three-dimensional flow. In: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia Canada
3. Beachkofski BK, Grandhi RV (2002) Improved Distributed Hypercube Sampling. In: 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO; USA
4. Bhattacharya M, Lu G (2003) A dynamic approximate fitness based hybrid ea for optimization problems. In: *Proceedings of IEEE Congress on Evolutionary Computation*, pp 1879–1886
5. Bishop CM (1995) *Neural Networks for Pattern Recognition*. Oxford University Press, UK
6. Bueche D, Schraudolph N, Koumoutsakos P (2005) Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics: Part C* 35(2):183–194
7. Ioannis C Kampolis, Kyriakos C Giannakoglou (2008) A multilevel approach to single- and multiobjective aerodynamic optimization. *Computer Methods in Applied Mechanics and Engineering* 197:2963–2975

8. Chen JH, Goldberg D, Ho SY, Sastry K (2002) Fitness inheritance in multi-objective optimization. In: Proceedings of Genetic and Evolutionary Computation Conference, Morgan Kaufmann
9. Chiba K, Obayashi S (2007) Data mining for multidisciplinary design space of regional-jet wing. *AIAA Journal of Aerospace Computing, Information, and Communication* 4(11):1019–1036, DOI: 10.2514/1.19404
10. Chiba K, Obayashi S, Nakahashi K, Morino H (2005) High-Fidelity Multidisciplinary Design Optimization of Wing Shape for Regional Jet Aircraft. In: Coello Coello CA, Hernández Aguirre A, Zitzler E (eds) *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, Springer. Lecture Notes in Computer Science Vol. 3410, Guanajuato, México, pp 621–635
11. Choi S, Alonso JJ, Chung HS (2004) Design of a low-boom supersonic business jet using evolutionary algorithms and an adaptive unstructured mesh method. In: *AIAA Paper 2004-1758*, 45th AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics and Materials Conference, Palm Springs, CA, USA
12. Chung HS, Alonso JJ (2004) Multiobjective optimization using approximation model-based genetic algorithms. In: *AIAA Paper 2004-4325*, 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Albany, New York, USA
13. Chung HS, Choi S, Alonso JJ (2003) Supersonic business jet design using a knowledge-based genetic algorithm with an adaptive, unstructured grid methodology. In: *AIAA Paper 2003-3791*, 21st Applied Aerodynamics Conference, Orlando, Florida, USA
14. Cinnella P, Congedo PM (2006) Optimal Airfoil Shapes for Viscous Transonic Flows of Dense Gases. In: *AIAA Paper 2006-3881*, 36th AIAA Fluid Dynamics Conference and Exhibit, San Francisco, California, USA
15. Coello Coello CA, Toscano Pulido G (2001) A Micro-Genetic Algorithm for Multiobjective Optimization. In: Zitzler E, Deb K, Thiele L, Coello CAC, Corne D (eds) *First International Conference on Evolutionary Multi-Criterion Optimization*, Springer-Verlag. Lecture Notes in Computer Science No. 1993, pp 126–140
16. Coello Coello CA, Toscano Pulido G (2001) Multiobjective Optimization using a Micro-Genetic Algorithm. In: Spector L, Goodman ED, Wu A, Langdon W, Voigt HM, Gen M, Sen S, Dorigo M, Pezeshk S, Garzon MH, Burke E (eds) *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, Morgan Kaufmann Publishers, San Francisco, California, pp 274–282
17. Coello Coello CA, Lamont GB, Van Veldhuizen DA (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edn. Springer, New York, ISBN 978-0-387-33254-3
18. Congedo PM, Corre C, Cinnella P (2007) Airfoil Shape Optimization for Transonic Flows of Bethe-Zel'dovich-Thompson Fluids. *AIAA Journal* 45(6):1303–1316, DOI: 10.2514/1.21615

19. Costa M, Minisci E (2003) MOPED: A Multi-objective Parzen-Based Estimation of Distribution Algorithm for Continuous Problems. In: Fonseca CM, Fleming PJ, Zitzler E, Deb K, Thiele L (eds) Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, pp 282–294
20. D’Angelo S, Minisci EA (2005) Multi-objective evolutionary optimization of subsonic airfoils by kriging approximation and evolutionary control. In: 2005 IEEE Congress on Evolutionary Computation (CEC’2005), Edinburgh, Scotland, vol 2, pp 1262–1267
21. Deb K (2001) Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester, UK, ISBN 0-471-87339-X
22. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197
23. Ducheyne EI, De Baets B, De Wulf R (2003) Is Fitness Inheritance Useful for Real-World Applications? In: Fonseca CM, Fleming PJ, Zitzler E, Deb K, Thiele L (eds) Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Springer. Lecture Notes in Computer Science. Volume 2632, Faro, Portugal, pp 31–42
24. Emmerich M, Giotis A, Özdenir M, Bäck T, Giannakoglou K (2002) Metamodel-assisted evolution strategies. In: Parallel Problem Solving from Nature, Springer, no. 2439 in Lecture Notes in Computer Science, pp 371–380
25. Fletcher R (1989) Practical Methods of Optimization. John Wiley and Sons, New York
26. Fonseca CM, Fleming PJ (1993) Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Forrest S (ed) Proceedings of the Fifth International Conference on Genetic Algorithms, University of Illinois at Urbana-Champaign, Morgan Kaufman Publishers, San Mateo, California, pp 416–423
27. Giunta A, Watson L (1998) A comparison of approximation modeling techniques: Polynomial versus interpolating models. Tech. Rep. 98-4758, AIAA
28. Glaz B, Goel T, Liu L, Friedmann PP, Haftka RT (2007) Application of a Weighted Average Surrogate Approach to Helicopter Rotor Blade Vibration. In: AIAA Paper 2007-1898, 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Honolulu, Hawaii, USA
29. Tushar Goel, Rajkumar Vaidyanathan, Raphael T Haftka, Wei Shyy, Nestor V Queipo, Kevin Tucker (2007) Response Surface Approximation of Pareto Optimal Front in Multi-Objective Optimization. *Computer Methods in Applied Mechanics and Engineering* 196:879–893

30. Goel T, Vaidyanathan R, Haftka R, Shyy W, Queipo N, Tucker K (2004) Response surface approximation of pareto optimal front in multiobjective optimization. Tech. Rep. 2004-4501, AIAA
31. Gonzalez LF, Périaux J, Srinivas K, Whitney EJ (2006) A generic framework for the design optimisation of multidisciplinary uav intelligent systems using evolutionary computing. In: AIAA Paper 2006-1475, 44th AIAA Aerospace Science Meeting and Exhibit, Reno, Nevada, USA
32. Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. *J Geophys res* 76:1905–1915
33. Hong YS, HLee, Tahk MJ (2003) Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks. *Engineering Optimization* 35(1):91–102
34. Hüscken M, Jin Y, Sendhoff B (2005) Structure optimization of neural networks for aerodynamic optimization. *Soft Computing* 9(1):21–28
35. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31(3):264–323, DOI <http://doi.acm.org/10.1145/331499.331504>
36. Jensen MT (2003) Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms. *IEEE Transactions on Evolutionary Computation* 7(5):503–515
37. Jeong S, Minemura Y, Obayashi S (2006) Optimization of combustion chamber for diesel engine using kriging model. *Journal of Fluid Science and Technology* 1:138–146
38. Jeong S, Suzuki K, Obayashi S, Kurita M (2007) Improvement of nonlinear lateral characteristics of lifting-body type reentry vehicle using optimization algorithm. In: AIAA Paper 2007–2893, AIAA infotech@Aerospace 2007 Conference and Exhibit, Rohnert Park, California, USA
39. Jin R, Chen W, Simpson T (2000) Comparative studies of metamodeling techniques under multiple modeling criteria. Tech. Rep. 2000-4801, AIAA
40. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing* 9(1):3–12
41. Jone D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box function. *Journal of Global Optimization* 13:455–492
42. Knowles J (2006) ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* 10(1):50–66
43. Knowles J, Nakayama H (2008) Meta-modeling in multiobjective optimization. In: Branke J, Deb K, Miettinen K, Slowinski R (eds) *Multiobjective Optimization-Interactive and Evolutionary Approaches*, Springer, pp 245–284
44. Kumano T, Jeong S, Obayashi S, Ito Y, Hatanaka K, Morino H (2006) Multidisciplinary design optimization of wing shape for a small jet aircraft using kriging model. In: AIAA Paper 2006-932, 44th AIAA Aerospace Science Meeting and Exhibit, Reno, Nevada, USA

45. Kung H, Luccio F, Preparata F (1975) On finding the maxima of a set of vectors. *Journal of the Association for Computing Machinery* 22(4):469–476
46. Landa Becerra R, Coello Coello CA (2006) Solving Hard Multiobjective Optimization Problems Using ε -Constraint with Cultured Differential Evolution. In: Runarsson TP, Beyer HG, Burke E, Merelo-Guervós JJ, Whitley LD, Yao X (eds) *Parallel Problem Solving from Nature - PPSN IX*, 9th International Conference, Springer. *Lecture Notes in Computer Science* Vol. 4193, Reykjavik, Iceland, pp 543–552
47. Langer H, Pühlhofer T, Baier H (2004) A multi-objective evolutionary algorithm with integrated response surface functionalities for configuration optimization with discrete variables. In: *AIAA Paper 2004-4326*, 10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference, Albany, New York, USA
48. Lee D, Gonzalez L, Periaux J, Srinivas K (2008) Robust design optimisation using multi-objective evolutionary algorithms. *Computer & Fluids* 37:565–583
49. Lee D, Gonzalez L, Srinivas K, Periaux J (2008) Robust evolutionary algorithms for uav/ucav aerodynamic and rcs design optimisation. *Computer & Fluids* 37:547–564
50. Lee DS, Gonzalez LF, Srinivas K, Auld DJ, Wong KC (2006) Aerodynamics/rcs shape optimisation of unmanned aerial vehicles using hierarchical asynchronous parallel evolutionary algorithms. In: *AIAA Paper 2006-3331*, 24th AIAA Applied Aerodynamics Conference, San Francisco, California, USA
51. Lee DS, Gonzalez LF, Srinivas K, Periaux J (2007) Multi-objective robust design optimisation using hierarchical asynchronous parallel evolutionary algorithms. In: *AIAA Paper 2007-1169*, 45th AIAA Aerospace Science Meeting and Exhibit, Reno, Nevada, USA
52. Lian Y, Liou MS (2004) Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithm. In: *AIAA Paper 2004-4323*, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, USA
53. Lian Y, Liou MS (2005) Multi-Objective Optimization of a Transonic Compressor Blade Using Evolutionary Algorithm. In: *AIAA Paper 2005-1816*, 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, Austin, Texas, USA
54. Lian Y, Liou MS (2005) Multi-objective Optimization of Transonic Compressor Blade Using Evolutionary Algorithm. *Journal of Propulsion and Power* 21(6):979–987
55. Mack Y, Goel T, Shyy W, Haftka R, Queipo N (2005) Multiple Surrogates for the Shape Optimization of Bluff Body-Facilitated Mixing. In: *AIAA Paper 2005-333*, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, USA

56. Nakayama H, Sawaragi Y (1984) Satisficing trade-off method for multi-objective programming. In: Grauer M, Wierzbicki A (eds) *Interactive Decision Analysis*, Springer, Heidelberg, pp 113–122
57. Nakayama H, Yun Y (2006) Support vector regression based on goal programming and multi-objective programming. In: *IEEE World Congress on Computational Intelligence*
58. Nakayama H, Inoue K, Yoshimori Y (2006) Approximate optimization using computational intelligence and its application to reinforcement of cable-stayed bridges. In: Zha X, Howlett R (eds) *Integrated Intelligent Systems for Engineering Design*, IOS Press, Amsterdam, pp 289 – 304
59. Obayashi S, Sasaki D (2002) Self-organizing map of pareto solutions obtained from multiobjective supersonic wing design. In: *AIAA Paper 2002–0991, 40th Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA
60. Ong YS, Nair PB, Keane AJ, Wong KW (2004) Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. In: Jin Y (ed) *Knowledge Incorporation in Evolutionary Computation, Studies in Fuzziness and Soft Computing*, Springer, pp 307–332
61. Pagano A, Federico L, Barbarino M, GUida F, Aversano M (2008) Multi-objective Aeroacoustic Optimization of an Aircraft Propeller. In: *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia Canada
62. Pierret S (1999) Turbomachinery blade design using a Navier-Stokes solver and artificial neural network. *ASME Journal of Turbomachinery* 121(3):326–332
63. Rai MM (2004) Robust Optimal Aerodynamic Design Using Evolutionary Methods and Neural Networks. In: *AIAA Paper 2004-778, 42nd AIAA Aerospace Science Meeting and Exhibit*, Reno, Nevada, USA
64. Rai MM (2004) Robust Optimal Design With Differential Evolution. In: *AIAA Paper 2004-4588, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, USA
65. Rasheed K, Ni X, Vattam S (2005) Comparison of methods for developing dynamic reduced models for design optimization. *Soft Computing* 9(1):29–37
66. Ratle A (1998) Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. In: Eiben A, Bäck T, Schoenauer M, Schwefel HP (eds) *Parallel Problem Solving from Nature*, vol V, pp 87–96
67. Reyes Sierra M, Coello Coello CA (2005) A Study of Fitness Inheritance and Approximation Techniques for Multi-Objective Particle Swarm Optimization. In: *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, IEEE Service Center, Edinburgh, Scotland, vol 1, pp 65–72
68. Sacks J, Welch W, Mitchell T, Wynn H (1989) Design and analysis of computer experiments (with discussion). In: *Statistical Science*, vol 4, pp 409 – 435

69. Sasaki D, Obayashi S, Nakahashi K (2001) Navier stokes optimization of supersonic wings with four objectives using evolutionary algorithms. In: AIAA Paper 2001-2531, 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, USA
70. Sasaki D, Obayashi S, Nakahashi K (2002) Navier-Stokes Optimization of Supersonic Wings with Four Objectives Using Evolutionary Algorithms. *Journal of Aircraft* 39(4):621-629
71. Schoenauer KAM (2002) Surrogate deterministic mutation. In: *Artificial Evolution'01*, Springer, pp 103-115
72. Smith RE, Dike BA, Stegmann SA (1995) Fitness inheritance in genetic algorithms. In: *SAC '95: Proceedings of the 1995 ACM symposium on Applied computing*, ACM Press, New York, NY, USA, pp 345-350
73. Song W, Keane AJ (2007) Surrogate-based aerodynamic shape optimization of a civil aircraft engine nacelle. *AIAA Journal* 45(10):265-2574
74. Srinivas N, Deb K (1994) Multiobjective Optimization Using Non-dominated Sorting in Genetic Algorithms. *Evolutionary Computation* 2(3):221-248
75. Todoroki A, Sekishiro M (2007) Dimensions and laminates optimization of hat-stiffened composite panel with buckling load constraint using multi-objective ga. In: *AIAA Paper 2007-2880, AIAA infotech@Aerospace 2007 Conference and Exhibit*, Rohnert Park, California, USA
76. Todoroki A, Sekishiro M (2008) Modified efficient global optimization for a hat-stiffened composite panel with buckling constraint. *AIAA Journal* 46(9):2257-2264
77. Ulmer H, Streicher F, Zell A (2003) Model-assisted steady-state evolution strategies. In: *Proceedings of Genetic and Evolutionary Computation Conference, LNCS 2723*, pp 610-621
78. Ulmer H, Streichert F, Zell A (2003) Evolution strategies assisted by gaussian processes with improved pre-selection criterion. In: *Proceedings of IEEE Congress on Evolutionary Computation*, pp 692-699
79. Vapnik V (1998) *Statistical Learning Theory*. Wiley
80. Vapnik VN (1995) *The Nature of Statistical Learning*. Springer
81. Voutchkov I, Keane AJ, Fox R (2006) Robust structural design of a simplified jet engine model, using multiobjective optimization. In: *AIAA Paper 2006-7003, Portsmouth, Virginia, USA*
82. Williams CKI, Rasmussen CE (1996) Gaussian processes for regression. In: *Touretzky DS, Mozer MC, Hasselmo ME (eds) Advances in Neural Information Processing Systems 8*, MIT Press
83. Won K, Ray T (2004) Performance of kriging and cokriging based surrogate models within the unified framework for surrogate assisted optimization. In: *Congress on Evolutionary Computation, IEEE*, pp 1577-1585
84. Zitzler E, Deb K, Thiele L (2000) Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8(2):173-195

85. Zitzler E, Laumanns M, Thiele L (2002) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: Giannakoglou K, Tsahalis D, Periaux J, Papailou P, Fogarty T (eds) EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, Athens, Greece, pp 95–100