Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

Executive summary

Meta-modeling and multi-objective optimization in aircraft design

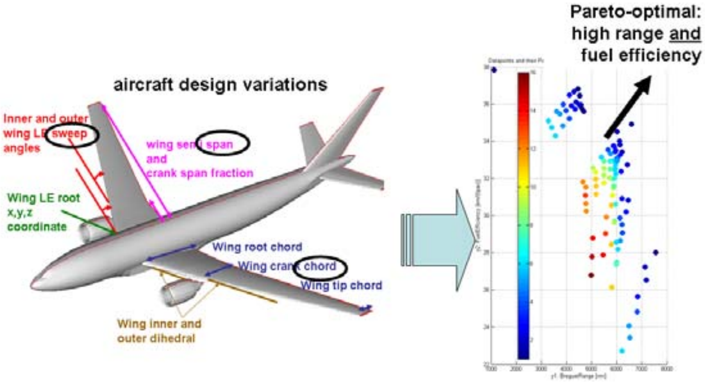

aircraft design variations
Inner and outer wing LE sweep angles
wing semi span and crank span fraction
Wing LE root x,y,z coordinate
Wing root chord
Wing crank chord
Wing tip chord
Wing inner and outer dihedral
Pareto-optimal: high range and fuel efficiency
y1: BreguetRange [nm]

Illustration of multi-objective aircraft design optimisation

Problem area

The global market and international competition are urging aeronautic industry to continuously enhance its performance in the engineering design process. The application of novel design methods and more accurate design criteria is of key importance for the further reduction of design time and increased design confidence level. The aerospace industry is also faced with a rapidly changing landscape driven by increasingly demanding environmental rules, consolidation in the whole industry, customer and competitive pressures
and consequently the emergence of new forms of doing business. These changes will influence the future aircraft product design process, requiring quicker, cheaper and more customised design and engineering processes.

Description of work

In the early phases of the aircraft design process, the design analyses are mostly based on relatively simple models and semi-empirical rules. Although computationally efficient, these analyses often have a limited range of validity, accuracy and flexibility. Therefore these methods are gradually being replaced by the more generic 'geometry and physics based' detailed design analysis methods

Report no.
NLR-TP-2009-718

Author(s)
W.J. Vankan
W.F. Lammen
R. Maas

Report classification
UNCLASSIFIED

Date
September 2010

Knowledge area(s)
Aerospace Collaborative Engineering and Design

Descriptor(s)
surrogate modelling
kriging models
genetic algorithms
multi-disciplinary analysis
aircraft wing design

This report is based on a book chapter (chapter 6) published in "Advances in Collaborative Civil Aeronautical Multidisciplinary Design Optimization", 2010, by AIAA.

that are generally applicable and potentially highly accurate, but on the other hand, mostly are computationally expensive. The computational cost of these accurate but expensive methods may become unacceptably high, in particular when used within automated design optimization loops that may require very many design analysis evaluations. In these optimization loops in the early aircraft design phases it is desirable to take into account multiple product properties of interest and optimize those simultaneously. Such simultaneous optimization can be effectively handled by multi-objective optimization algorithms, which allow for a more general optimization problem formulation, but also need more objective function evaluations in comparison to traditional single-objective optimization algorithms.

Results and conclusions

This document presents a methodology in which various advanced interpolation and approximation techniques and optimization algorithms are applied in a meta-model based optimization approach for aircraft design problems. The key of the approach lies in the de-coupling of the (computationally expensive) aircraft design analyses from the automated search and optimization process. The search process now makes use of the compact and computationally efficient meta-model and allows for high flexibility for further investigations. The meta-model, in turn, is based on a set of results that are obtained with the computationally expensive aircraft design analyses. This optimization approach is applied to the multi-disciplinary design and multi-objective optimization of aircraft wings. In this case study, the wing planform and the maximum take-off weight of a generic transonic aircraft are optimized where the aircraft range and fuel consumption are the objective functions. The case study shows that the methodology described in this document provides an efficient procedure for calculating the Pareto optimal design points of the multi-objective design optimization.

Applicability

The combination of advanced meta-models and multi-objective optimization algorithms for aircraft design presented in this document is flexible and applicable to a variety of design processes. The improved efficiency and flexibility of the design process contributes to the quicker, cheaper and more customised design and engineering processes for aircraft design. Dutch aeronautic industry can apply these improved engineering processes, as such ensuring their place in the supply chain of leading aircraft manufacturers, both technologically and commercially.

Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2009-718

Meta-modeling and multi-objective optimization in aircraft design

W.J. Vankan, W.F. Lammen and R. Maas

This report is based on a book chapter (chapter 6) published in "Advances in Collaborative Civil Aeronautical Multidisciplinary Design Optimization", 2010, by AIAA.

This publication has been refereed by the Advisory Committee AEROSPACE VEHICLES.

Customer	National Aerospace Laboratory NLR
Contract number	----
Owner	National Aerospace Laboratory NLR
Division NLR	Aerospace Vehicles
Distribution	Unlimited
Classification of title	Unclassified
	September 2010

Approved by:

Author	Reviewer	Managing department
27/9/2010	27/9/2010	27/09 2010.

Chapter 6

Meta-modeling and multi-objective optimization in aircraft design

W.J. Vankan, W. Lammen, R. Maas
National Aerospace Laboratory NLR, Amsterdam, the Netherlands

Table of contents

Abbreviations

AAE	Average Absolute Error
ANN	Artificial Neural Networks
BLUP	Best Linear Unbiased Predictor
CFD	Computational Fluid Dynamics
DOE	Design Of Experiments
FEM	Finite Element Method
GA	Genetic Algorithm
kriging-cC	kriging-constant-Cubic fit (kcc)
kriging-cE	kriging-constant-Exponential fit (kce)
kriging-cG	kriging-constant-Gauss fit (kcg)
kriging-lC	kriging- linear-Cubic fit (klc)
kriging-lE	kriging- linear-Exponential fit (kle)
kriging-lG	kriging-linear-Gauss fit (klg)
LoD	Lift over Drag (L/D)
MAE	Maximum Absolute Error
MAPE	Mean Absolute Percentage Error
MARS	Multivariate Adaptive Regression Splines
MDA	Multidisciplinary Design Analysis
MDO	Multidisciplinary Design and Optimization
MNSGA	Non-dominated Sorting Genetic Algorithm in Matlab

MOO	Multi-Objective Optimization
MTOW	Maximum Take-Off Weight
NSGA	Non Dominated Sorting Genetic Algorithm
poly0	0th order polynomial fit
poly1	1st order polynomial fit
poly2	2nd order polynomial fit
poly3	3rd order polynomial fit
poly4	4th order polynomial fit
poly5	5th order polynomial fit
RBF	Radial Basis Functions
RMSE	Root Mean Squared Error
SBX	Simulated Binary Crossover
SVR	Support Vector Regression

Abstract

Design and analysis methods for multidisciplinary design and multi-objective optimization of aircraft are continuously improving in accuracy and reliability. Correspondingly, the increased computational complexity leads to high costs in terms of time, effort and money, needed for these analyses. In order to drastically reduce these costs, so-called meta-models based on fitting methods can be used.

This chapter presents methodologies in which various advanced interpolation and approximation techniques and optimization algorithms are applied in a meta-model based optimization approach for aircraft design problems. These methodologies are demonstrated in a multi-objective optimization study of aircraft design in terms of range and fuel consumption.

The results demonstrate the flexibility and the potential of these methodologies by tackling a complex design optimization problem at a relatively low computational cost and sufficient accuracy of the meta-models applied.

Because of their computational efficiency, the meta-modeling methodologies allow for a significant time- and cost-effective assessment of high-dimensional design problems involving large scale

computational analyses, as is illustrated by the aircraft wing optimization study. These methodologies have also been applied in an engine turbine disk design case and a helicopter pre-design case.

I Introduction – MDA, MDO and MOO in aeronautics

The continuous development of the methodologies for aircraft design and analysis is aimed at achieving higher levels of detail in shorter analysis turn around cycles. Moreover, driven by ever increasing technical and commercial requirements due to global competition, more detailed design analyses are being required and applied in earlier phases of the aircraft design process where there are still very many degrees of freedom and few restrictions on the design space[1]. The analyses traditionally used in the early phases of the aircraft design are mostly based on semi-empirical rules[2]. Although computationally efficient, these analyses often have a limited range of validity, accuracy and flexibility. Therefore these methods are gradually being replaced by the more generic 'geometry and physics based' detailed design analysis methods that are generally applicable and potentially highly accurate[3]. However, these methods are mostly computationally expensive.

Also, the required design analyses in aircraft multidisciplinary design and optimization (MDO) need to comprise case-dependent sets of disciplines and may be difficult to combine into an integrated aircraft design system. Therefore such integrated aircraft design system is usually developed for a specific range of design problems, such as optimization of blended-wing-body[4, 5] or transonic transport aircraft planform[6]. Moreover, such integrated aircraft design system often requires specific software (e.g. particular analysis tools) and hardware (e.g. dedicated computer servers), and is therefore prone to operational issues such as temporal unavailability of servers or licenses. The computational cost of (some of) the analyses in the integrated aircraft design system is another issue to be handled, especially when used within automated search or optimization loops that typically may require many (e.g. thousands) design analysis evaluations.

In order to deal with the above mentioned issues, various approximation and interpolation methods have been proposed. These methods have shown to effectively deal with the issues[7, 8, 9] by providing compact, accurate and computationally efficient representations of the considered properties of the underlying aircraft design (in optimization context also termed as design objectives or fitnesses). In this chapter the term *meta-models* will be used to identify the above mentioned methods; in literature also alternative terms like surrogate models or response surface models are used. The key of the approach presented in this chapter lies in the de-coupling of the (computationally expensive) aircraft design analyses from the automated search and optimization process. The search process now makes use of the compact and computationally efficient meta-model and allows for high flexibility for further investigations. The meta-model, in turn, is based on a set of results that are obtained with the computationally expensive integrated aircraft design analyses. For this purpose, effective sampling of the multi-dimensional design domain is achieved by Design Of Experiments (DOE) methods. In the selected sample points the design objectives and constraints are evaluated by (parallel) computations with integrated aircraft design analysis systems.

Many different fitting methods are available for the creation of the meta-models (e.g.[7], each with different advantages for different types of problems. In this chapter we consider a number of different fitting (interpolation and approximation) methods, and compare the meta-model quality for these fitting methods on the basis of appropriately defined quality-of-fit criteria. A proprietary multi-dimensional and multi-method data fitting software tool (MultiFit) [10] is used to statistically analyze the data sets that result from the design evaluations and to generate meta-models using different fitting methods. The representativeness of the meta-models is investigated and the most suitable meta-models are applied in the aircraft design process where several optimization algorithms are used to find the most promising aircraft designs.

Traditionally, optimization methods are applied to automatically search for design variations by which one property, expressed in an objective function, is optimized subject to certain constraints. However, in aircraft design it is desirable to take into account multiple product properties of interest and optimize those simultaneously. Typical properties in aircraft design that need to be optimized are for example weight, lift, drag, payload, range, etc. (e.g. [11, 12, 13 and 14]). Such simultaneous optimization can be expressed by a multi-objective optimization (MOO) problem formulation, which can be considered as a generalization of the traditional (single objective) optimization problem[15]. Besides being more general, MOO problems are also more costly to solve (in terms of numbers of calculations, i.e. objective function evaluations) and may become quite complex in cases of high dimensional optimization problems. A multitude of methods is available for dealing with MOO problems[16, 17, 18, 19, 20, 21, 22, 23], of which many have evolved quite recently and are still under significant development. Among these methods, the category of evolutionary algorithms has proven to be quite efficient for MOO problems. Some examples of evolutionary algorithms are genetic algorithms, evolutionary computing, evolution strategies and evolutionary programming, of which some will be described in more detail in the following sections.

In this document the methodology for creating and assessing the meta-models is described. This methodology is based on a combination of various existing techniques for multi-dimensional data fitting and statistical assessment. It is aimed at the efficient application of many different fitting methods to multi-dimensional data sets, as such extending on the commonly used tools in this area which are often limited in dimensionality or number of available fitting methods. The different fitting methods that are considered here are described in the next sections, as well as the way in which the quality-of-fit criteria are determined. Also, some state-of-the-art methods for multi-objective optimization are described. For illustration of the approach and its benefits, an aircraft design optimization case study is described. Here, the meta-modeling approach is applied to the multi-disciplinary design and multi-objective optimization of aircraft wings. In this case study, the wing planform and the maximum take-off weight of a generic transonic aircraft are optimized where the aircraft range and fuel consumption are the objective functions. More information on the multidisciplinary design analysis (MDA) system is provided in [40], chapter 5 (wing MDO).

II Meta-modeling

In aircraft design problems the aim is to improve or optimize the characteristics (design objectives such as performance, behavior, etc.) of the product by variation of its properties (design parameters such as shape, material, etc.). In general the product's properties (denoted as a vector **x** in this chapter) and characteristics (denoted as a vector **y** in this chapter) are expressed as real-valued (continuous) quantities and their inter-dependency (denoted as a vector function **f** in this chapter) is non-linear ($\mathbf{y}=\mathbf{f}(\mathbf{x})$). Evaluation of the function **f** is often costly (in terms of time and computer resources) and may involve (iterative) computational analyses (e.g. finite element or computational fluid dynamics). Because of these computational complications it is desirable to retrieve efficiently, i.e., using as few as possible function evaluations, the desired product's characteristics (**y**) in the considered design domain (i.e. for the set of allowable values of the product's properties **x**). This may be achieved, for example, by direct optimization of **y** for **x** using efficient gradient based optimization algorithms (e.g. [24]).

However, lack of accurate gradient or Jacobian information (d**y**/d**x**), limited robustness and reliability of the computational analyses, or convergence into local sub-optima, may hamper the effectiveness of this approach. It is therefore beneficial to apply 'gradient-free' global search methods, such as genetic algorithms and pattern search (e.g. [26]), besides the gradient based optimization algorithms. The large number of evaluations of the objective function (**y**=**f**(**x**)) that are typically needed by these search methods do not allow for a high computational cost per evaluation. An efficient approximate representation (**y***=**f***(**x**)), also-called meta-model, of the design problem is therefore required.

A variety of fitting methods, such as polynomial regression, neural networks, and kriging models, are available for creating such meta-models[7] from sampled data sets ($\mathbf{x}^i$, $\mathbf{y}^i$) of the design problem. In order to achieve an optimal meta-model the most suitable fitting method for the considered design problem should be applied. There exist various statistical verification and cross validation methods[26, 27] by which the quality (or 'representativeness') of the different meta-models can be assessed and the most suitable method can be identified. These methods consider a (small) subset of the data set, called verification points, in which the error of the prediction (**y**-**y***) is evaluated for a fit that is made for the data set without the verification points. For user-friendly creation, assessment and comparison of fits with a wide range of multi-dimensional interpolation and approximation methods, a dedicated software tool[10] was developed, which is used in the further evaluation and optimization of the considered design problem.

The two most used model classes in the analyses in this chapter are the polynomial models and the kriging model, which are described below. Besides these two, several other interpolation and approximation methods are available[10], such as Artificial Neural Networks (ANN) and Radial Basis Functions (RBF) models, of which a brief description is also given below. It should be noted here that kriging and RBF models are interpolating models, i.e. they are exact in the given data points, whereas polynomial models and ANN are approximating models.

A Polynomials

One commonly used and relatively simple class of meta-models are the polynomial regression models, which are also often referred to as response surface models. The coefficients of the polynomial regression model are usually determined according to a least-squares procedure, i.e. by minimizing the sum of the squared residuals (the difference between the values predicted by the polynomial model and values in the data set). Polynomial regression models provide a global representation of the data, i.e. consist of one regression function for the whole considered domain. Polynomial functions of different orders can be used as regression function. The well known linear regression model is the first order polynomial approximation to the data. Of course also a "zeroth order" regression model of the data can be made, which is equal to the mean value of the data. Higher order models can also be built. However, with higher orders polynomials the risk of so called over-fitting of the data increases. In this case the (high order) polynomial follows the data so closely such that it does captures local oscillations in the data, but it does not properly capture the global behavior of the data, as such compromising the quality of the prediction for points not in the data set. Furthermore the reliability of polynomial models for extrapolation (i.e. predicting output values for input values (slightly) outside the range of the available data) is in general very poor, in particular for high order polynomial models.

To illustrate the creation of polynomial regression models, consider the following simple example, in which we assume a data set that consists of N input-output combinations (data points) $\{(\mathbf{x}^i,y^i)|i=1,\ldots,N\}$, where the input is 3-dimensional $\mathbf{x}^i=(x^i_1,x^i_2.x^i_3)$ and the output y^i is 1-dimensional. The following second order polynomial regression model can be fitted to the data.

$$\begin{aligned} y^* &= c_0 + c_1x_1 + c_2x_2 + +c_3x_3 + c_4x_1x_2 + c_5x_1x_3 + c_6x_2x_3 \\ &+ c_7{x_1}^2 + c_8{x_2}^2 + c_9{x_3}^2 \end{aligned} \qquad 1$$

This model $y*$ is linear in the coefficients c_j and the optimal c_j are found by least squares:

$$\mathbf{c} = \begin{pmatrix} c_0 \\ \vdots \\ c_9 \end{pmatrix} = \left(A^T A\right)^{-1} A^T \mathbf{y}_{data} \text{ where}$$

$$A = \begin{pmatrix} 1 & x_1^1 & x_2^1 & x_3^1 & x_1^1 x_2^1 & x_1^1 x_3^1 & x_2^1 x_3^1 & {x_1^1}^2 & {x_2^1}^2 & {x_3^1}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^N & x_2^N & x_3^N & x_1^N x_2^N & x_1^N x_3^N & x_2^N x_3^N & {x_1^N}^2 & {x_2^N}^2 & {x_3^N}^2 \end{pmatrix} \text{ and} \tag{2}$$

$$\mathbf{y}_{data} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

where A is the so-called "design matrix" for the input data $\{\mathbf{x^i}\}$. For other input dimensions and other orders of the polynomial the same procedure can be applied. A polynomial of order m in a k-dimensional input space has $\begin{pmatrix} m+k \\ k \end{pmatrix}$ coefficients, which is then also the minimum number of the input-output combinations ($\mathbf{x}^i$,y^i) data set that are required for creating this polynomial model. In practice the minimum number of data points needed for an acceptable fit is about 1.5 the number of coefficients.

B Kriging

Kriging models have in the past decade evolved as a new and quite effective meta-modeling technique. Originating from geo-statistics, kriging models have found their way into modeling of data coming from computationally expensive simulations, often referred to as Design and Analysis of Computer Experiments (DACE). In[28], Lophaven and co-workers present a widely used Matlab implementation of the kriging method (the Matlab kriging toolbox DACE). While often the derivation of the kriging models is fully statistical, a more deterministic explanation is given in this chapter following some of the notions given in[28].

The kriging model combines a global regression model and local refinements into one interpolation model of a given data set. As will be explained below, the global model is a generalized regression model on top of which a local model is built using a correlation function to couple points in the data set that are in the vicinity of each other. (Illustrated in figure 1 below).

As explained in the previous section, polynomial functions can be fitted to a data set using a least squares approach. The underlying assumptions in the least squares fit (eq. 2) is that the residuals between the fit and the data are equally important in all data points and no correlation exists between the residuals in two data points. In kriging models this correlation between the residuals is assumed to exist given by the correlation function *g* defined as:

$$g(\boldsymbol{\theta},\mathbf{x},\tilde{\mathbf{x}}) = g\big((\theta_1 \ldots,\theta_k),(x_1,\ldots,x_k),(\tilde{x}_1,\ldots,\tilde{x}_k)\big) = \prod_{j=1}^{k} g_j(\theta_j,x_j,\tilde{x}_j)\,,$$

Where θ is a scaling parameter for the correlation function, and *x* and $\tilde{x}$ are two different

input points, such that: 3

$$g(\boldsymbol{\theta},\mathbf{x},\tilde{\mathbf{x}}) \geq 0$$

and

$$g(\boldsymbol{\theta},\mathbf{x},\mathbf{x}) = 1$$

An example of such a function is a Gaussian correlation function:

$$g\big((\theta_1 \ldots,\theta_k),(x_1,\ldots,x_k),(\tilde{x}_1,\ldots,\tilde{x}_k)\big) = \prod_{j=1}^{k} e^{-\theta_j (x_j-\tilde{x}_j)^2} \qquad 4$$

The regression part of the kriging model is now obtained by regression model to the data points with a low order polynomial (up to second order) taking the correlation relations into account. The best fit to the data is obtained by the so-called generalized least squares fit to the data:

$$\mathbf{c} = \begin{pmatrix} c_0 \\ \vdots \\ c_9 \end{pmatrix} = \left(A^T R^{-1} A\right)^{-1} A^T R^{-1} \mathbf{y}_{data}\ , \qquad 5$$

where *A* again is the polynomial design matrix of *x*, as in eq. 2, and $\boldsymbol{y}_{\text{data}}$ is the vector containing the output values of all data points. *R is* the NxN correlation matrix of the data points $\mathbf{x}^1,\ldots,\mathbf{x}^n$:

$$R = \begin{pmatrix} g(\boldsymbol{\theta}, \mathbf{x}^1, \mathbf{x}^1) & g(\boldsymbol{\theta}, \mathbf{x}^1, \mathbf{x}^2) & \cdots & g(\boldsymbol{\theta}, \mathbf{x}^1, \mathbf{x}^n) \\ g(\boldsymbol{\theta}, \mathbf{x}^2, \mathbf{x}^1) & g(\boldsymbol{\theta}, \mathbf{x}^2, \mathbf{x}^2) & \cdots & g(\boldsymbol{\theta}, \mathbf{x}^2, \mathbf{x}^n) \\ \vdots & \vdots & \ddots & \vdots \\ g(\boldsymbol{\theta}, \mathbf{x}^n, \mathbf{x}^1) & g(\boldsymbol{\theta}, \mathbf{x}^n, \mathbf{x}^2) & \cdots & g(\boldsymbol{\theta}, \mathbf{x}^n, \mathbf{x}^n) \end{pmatrix} \qquad 6$$

The kriging model consists of this regression model plus an correction model based on the correlation function such that the model is interpolating. For a quadratic regression function and a correlation function g the kriging model can be written as (again for the 3-dimensional example):

$$y^*(\mathbf{x}) = \begin{pmatrix} 1 & x_1 & x_2 & x_3 & x_1x_2 & x_1x_3 & x_2x_3 & x_1^2 & x_2^2 & x_3^2 \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_9 \end{pmatrix} + , \qquad 7$$
$$+ r(\mathbf{x})v(\mathbf{x})$$

where $r(\mathbf{x}) = \begin{pmatrix} g(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}^1) & \cdots & g(\boldsymbol{\theta}, \mathbf{x}, \mathbf{x}^n) \end{pmatrix}$ and r(**x**)v(**x**) make up the local correction to the regression model. v(**x**) is an n-dimensional vector that can be derived easily due to the fact that the kriging method as explained here is an interpolating method in the known data points. Combining equation 7 for all the known data points gives:

$$\mathbf{y}_{data} = A\mathbf{c} + Rv(x) \Rightarrow v(x) = R^{-1}\left(\mathbf{y}_{data} - A\mathbf{c}\right) \qquad 8$$

Thus for an arbitrary point x the kriging approximation becomes:

$$y(x)^* = a(x)\mathbf{c} + \mathrm{r}(\mathrm{x})\, R^{-1}\left(\mathbf{y}_{data} - a(x)\mathbf{c}\right), \text{ with } \mathbf{c} = \begin{pmatrix} c_0 \\ \vdots \\ c_9 \end{pmatrix} = \left(A^T R^{-1} A\right)^{-1} A^T R^{-1} \mathbf{y}_{data} \qquad 9$$

where the first term on the right hand side represents the global regression model and the second term represents the local correlation model.

In[28], three different regression models are considered: 0th (i.e., constant), 1st (i.e. linear) and 2nd (i.e. quadratic) order polynomials. If the regression is restricted to the constant function, then one speaks of ordinary kriging, while kriging with higher order polynomial regression is called universal kriging. Also several correlation functions may be applied. Besides the already mentioned Gaussian correlation function, also the exponential and the cubic spline correlation function are available[28].

For convenience, the formulations of the exponential and the cubic spline correlation function are also given below:

$$g\big((\theta_1 \ldots, \theta_k), (x_1, \ldots, x_k), (\tilde{x}_1, \ldots, \tilde{x}_k)\big) = \prod_{j=1}^{k} e^{-\theta_j |x_j - \tilde{x}_j|} \text{ (exponential correlation function)} \qquad 10$$

The cubic spline correlation function

$$g\big((\theta_1 \ldots, \theta_k), (x_1, \ldots, x_k), (\tilde{x}_1, \ldots, \tilde{x}_k)\big) = \prod_{j=1}^{k} g_j(\theta_j \mid x_j - \tilde{x}_j \mid)$$

with

$$g_j(v) = \begin{cases} 1 - 15v^2 + 30v^3 & v \le 0.2 \\ \frac{5}{4}(1-v)^3 & 0.2 < v < 1 \\ 0 & v \ge 1 \end{cases} \qquad 11$$

The correlation functions have scale parameters θ_j. Although kriging models yield interpolating fits through the data points, independent of the values for the θ_j parameters, the shape of the model does depend on these θ_j parameters. If all θ_j are large, then the overall model will be (approximately) the regression model with local spikes around the available data points, whereas for small θ_j the model will interpolate more smoothly through the data points. In the creation of the kriging models, optimal values for the θ_j are determined based on minimization of the log-likelihood formulation[28]

$$\theta_{opt} = \arg\min_{\theta}\left\{-\frac{1}{2}k\log\left(\left(\mathbf{y}_{data} - Ac\right)^{T} R^{-1}\left(\mathbf{y}_{data} - Ac\right)\right) + \log(\det(R))\right\} =$$

$$= \arg\min_{\theta}\left\{\left(\left(\mathbf{y}_{data} - Ac\right)^{T} R^{-1}\left(\mathbf{y}_{data} - Ac\right)\right)^{2} \det(R)^{\frac{1}{k}}\right\}$$

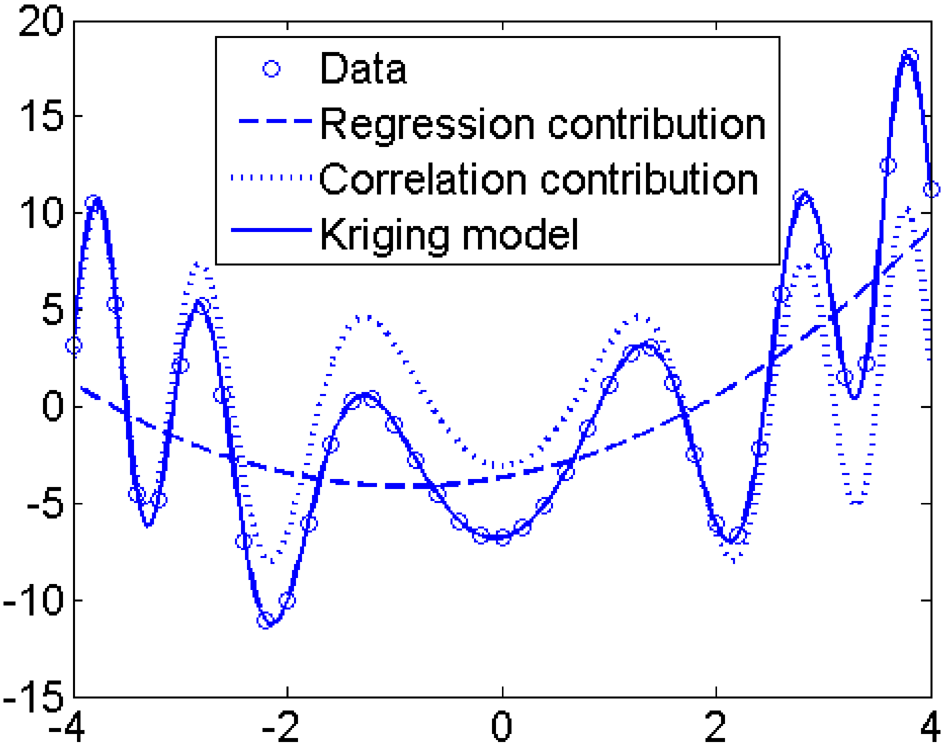


Fig. 1 Plot of a simple kriging approximation and the different contributions to the model

As a closing remark it should be noted that the kriging predictor has another interesting property. It can be proven that under certain restrictions a kriging model is a so-called Best Linear Unbiased Predictor (BLUP) in the data values y^i. This is generally the starting point of the explanation of the kriging models from a statistical point of view.

C Artificial Neural Networks and Radial Basis Functions

In this section, ANN and RBF models will only be briefly described. Further information about those methods can be found in[39]. In this description only feed-forward ANNs with back-propagation learning functions are considered. The chosen architecture consists of input nodes, one hidden layer and an output layer. The hidden layer has a tanh (tan-sigmoid) activation function, while the output layer has a linear activation function. The architecture of this type of ANN is explained in the Fig. 2.

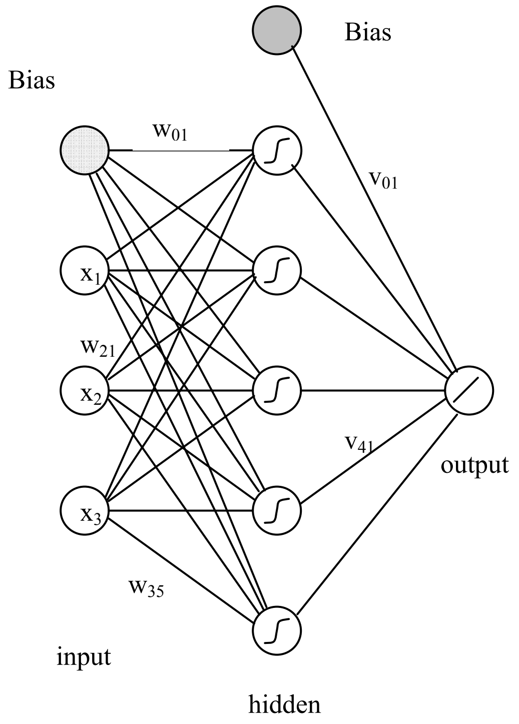


Fig. 2 Example of a feed-forward ANN. The inputs (including a constant bias) are multiplied by weights (w_{ij}) and summed at a hidden node after which the transfer function is applied. The resulting values from the hidden layer are multiplied by the weights v_{ij} and then summed to give the output of the network.

Due to the non-linear activation functions of the nodes in the hidden layer non-linear input-output relations can be modeled. However building an ANN, i.e., computing the optimal weights, requires a training for which in general many input-output combinations (i.e., data points) are needed. As a general rule of thumb, the minimum number of data points required is often computed as three times the number of connections in the network (i.e. $3\ (\ (n_{in}+1)n_h+(n_h+1)n_{out})$).

The Radial Basis Function (RBF) method considered here is an interpolating method on all data points. In this method so-called radial basis functions are centered on the input data points. Although the radial basis function can have (almost) any radial symmetric form, they are normally Gaussian functions:

$$f(\mathbf{x},\mathbf{c}) = e^{-\theta\|\mathbf{x}-\mathbf{c}\|^2} \qquad 13$$

where **c** is the centre of the radial basis function, $\|x-c\|$ is the (Euclidean) distance from **x** to and **c**, and θ is a parameter (chosen in advance) that determines the width of the radial basis function.

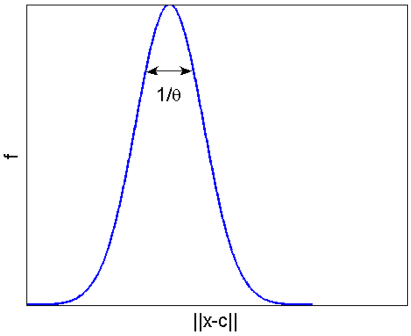


Fig. 3 Sketch of a radial basis function

The predictive model based on the data pairs $\{(\mathbf{x}^1,y^1),\ldots,(\mathbf{x}^n,y^n)\}$ is then:

$$y(\mathbf{x}) = \sum_{j=1}^{n} w_j e^{-\theta\|\mathbf{x}-\mathbf{x}^j\|^2} \text{ with } \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} e^{-\theta\|\mathbf{x}^1-\mathbf{x}^1\|^2} & \cdots & e^{-\theta\|\mathbf{x}^1-\mathbf{x}^n\|^2} \\ \vdots & \ddots & \vdots \\ e^{-\theta\|\mathbf{x}^n-\mathbf{x}^1\|^2} & \cdots & e^{-\theta\|\mathbf{x}^n-\mathbf{x}^n\|^2} \end{pmatrix}^{-1} \begin{pmatrix} y^1 \\ \vdots \\ y^n \end{pmatrix} \qquad 14$$

D Assessments of meta-models

Having so many different possibilities to build meta-models, tools to select the best model for the task at hand are needed. Some of the methods presented above (e.g. kriging models, RBF) are interpolating methods which by definition are exact in the data points used in the modeling, while the approximating methods (polynomials, ANN) are not. However it is not the exact value in the known points that is the most important, but the capability of the meta-model to generalize well over the total input space. For assessment of this capability there are different methods available[10]:

1. Validation by an independent data set that was not used in the creation of the model
2. Leave-one-out cross validation
3. P-fold cross validation

In the first method the available data is split into two sets. With the first set the meta-model is built, while the approximation error on the second set is considered. A drawback of this method is that the comparison of the methods is only done in (normally a few) selected points and thus this method has only local validity and is not very robust. On the other hand, the advantage is that the fitting and validation are only done once which results in a better efficiency.

In the second method, the leave-one-out cross validation, each single data point is used in turn for validation while a meta-model is built with the other N-1 data points. While now some global quality measure is obtained, the drawback of this method is that all considered fitting methods have to be applied and validated N times, which can be computationally expensive for large data sets.

The third method, the p-fold cross validation method, provides a global coverage of the estimated error while keeping computation time low. In this method the data set is split into p disjoint subsets of about equal size. While one subset is kept as validation set the models are built using the rest of the data. Notice that when p=N this methods is equal to the leave-one-out method. However when N is large a p-fold cross validation with a moderate p value can be a computationally efficient alternative to leave-one-out cross validation with regard to obtaining a global fit quality measure.

In order to assess the fit quality, several error metrics can be applied to the validation points, for example:

RMSE (Root Mean Square Error)	$\sqrt{\dfrac{\sum_{i}^{N_{val}} (y_i - y_i^*)^2}{N_{val}}}$

AAE (Average Absolute Error)	$\dfrac{\sum_{i}^{N_{val}} \left\| y_i - y_i^* \right\|}{N_{val}}$
MAE (Maximum Absolute Error)	$\max\limits_{i=\{1,\ldots N_{val}\}} \left\| y_i - y_i^* \right\|$
MAPE (Mean Absolute Percentage Error)	$\dfrac{\sum_{i}^{N_{val}} \left\| \frac{y_i - y_i^*}{y_i} \right\|}{N_{val}} \cdot 100\%$

The first three measures are absolute measures, quantifying different misfit properties of the model. The fourth error measure is a relative error measure. In case of leave-one-out or p-fold cross validation, the predictions of the validation points by all the different fits are taken together, and then the error formulas are applied.

E Illustration of fitting methods

To conclude this section, some examples of fits and error measures on a simple sine function are presented. The one dimensional data set consists of 11 data points (x_k, y_k) Є ($\{(\frac{2\pi}{10}k, \sin(\frac{2\pi}{10}k)) \mid k = 0,\ldots,10\}$). To these data points, four models are fitted:

- a second order polynomial (poly2)
- a third order polynomial (poly3)
- a kriging model with constant regression function and Gaussian correlation function (krigingcG)
- a kriging model with constant regression function and exponential correlation function (krigingcE)

The results are plotted in Fig. 4. The krigingcG model gives the best representation of the sine function, while the poly2 model based on this data set approximates the underlying sine function very poorly. To assess the quality of the different fits, 629 data points are generated with the sine function $\{(\frac{k}{100}, \sin(\frac{k}{100}) \mid k = 0,\ldots,628\}$. These 629 data points are then used as validation points and the four described error measures are computed for the four different models (Table 1). In Fig. 4 the results are presented, again indicating, according to each of the four error measures, that krigingcG is the best model in this case.

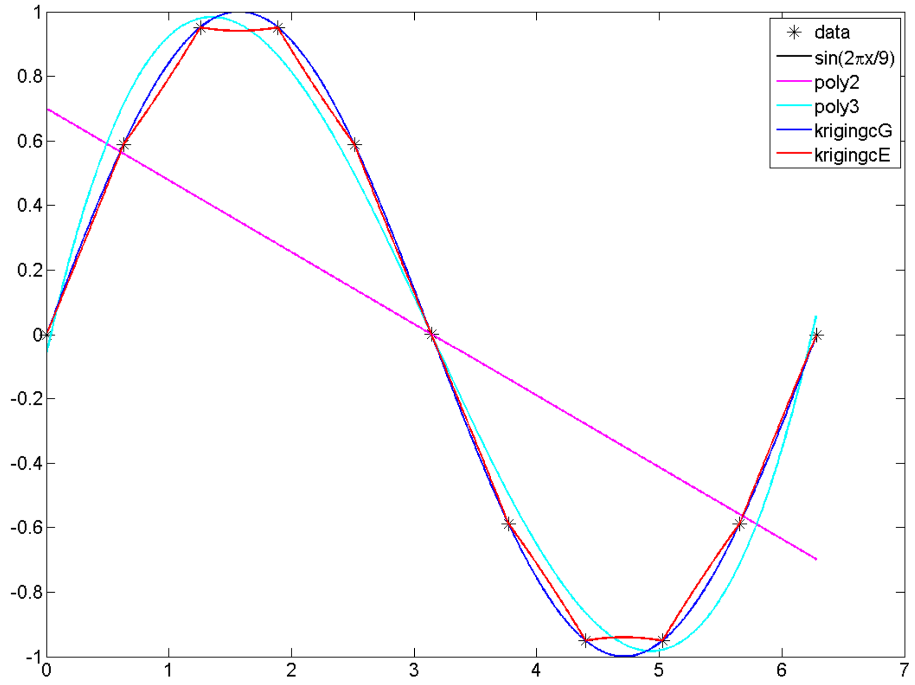


Fig. 4 Several models fitted to a period of the sine function

Table 1. Fit quality for various fitting functions

	poly2	poly3	krigingcG	krigingcE
RMSE	0.466995	0.079639	1.25E-06	0.030739
AAE	0.417659	0.071183	6.06E-07	0.025641
MAE	0.699474	0.124509	4.17E-06	0.059566
MAPE	184.6193	18.9017	0.159321	4.942767

.

III Optimization – MDA, MDO and MOO in aeronautics

Product design mostly aims for improvement (or optimization when possible) of one or more product properties. Traditionally, optimization methods (single objective) are applied to automatically search for design variations by which one property, expressed in an objective function, is optimized subject to certain constraints. However, it would be sensible to take into account all the product properties of interest and optimize those simultaneously. Typical properties of interest in aircraft design are for example weight, lift, drag, payload, range, etc. (e.g.[10, 11, 12, 13 and 14]). Such simultaneous optimization can be performed by multi-objective optimization (MOO) methods, which can be considered as a generalization of the traditional (single objective) optimization approach[15]. Besides being more general, MOO methods are also more costly (in terms of calculations, i.e. objective function evaluations) and may become quite complex in cases of high dimensional optimization problems.

A multitude of methods is available for dealing with MOO problems[16, 17, 18, 19, 20, 21, 22, 23], of which many have evolved quite recently and are still under significant development. Besides the traditional gradient based optimization algorithms, also another category of algorithms is considered, which can be characterized as evolutionary algorithms. Evolutionary algorithms can be considered as computer-based problem solving systems that use computational models of some of the known mechanisms of evolution as key elements in their design and implementation[29]. Some examples of different types of evolutionary algorithms are genetic algorithms, evolutionary computing, evolution strategies and evolutionary programming.

This section provides a description of the general multi-objective optimization problem formulation, and an overview of several state-of-the-art algorithms for solving these multi-objective optimization problems.

A Multi-objective optimization

Multi-objective optimization, as opposed to single-objective optimization, considers the optimization of vector-valued objective functions instead of single scalar-valued objective functions. In the case of single-objective optimization, comparing solutions in single objective space is trivial: we simply say (in case of a minimization problem) that a solution $\mathbf{x}^1 \in \mathbf{X}$ is **better** than another solution $\mathbf{x}^2 \in \mathbf{X}$ if $y^1 < y^2$, where $y^1 = f(\mathbf{x}^1)$ and $y^2 = f(\mathbf{x}^2)$.

As an illustration, consider the solution of the simple single-objective optimization problem:

$$\min_{x} \; y = \sin(x) \quad ; \quad x \in [0,2\pi],$$

with the solution $\mathbf{x}^1 = \frac{3\pi}{2}$ with objective value $y^1 = -1$ (Fig. 5).

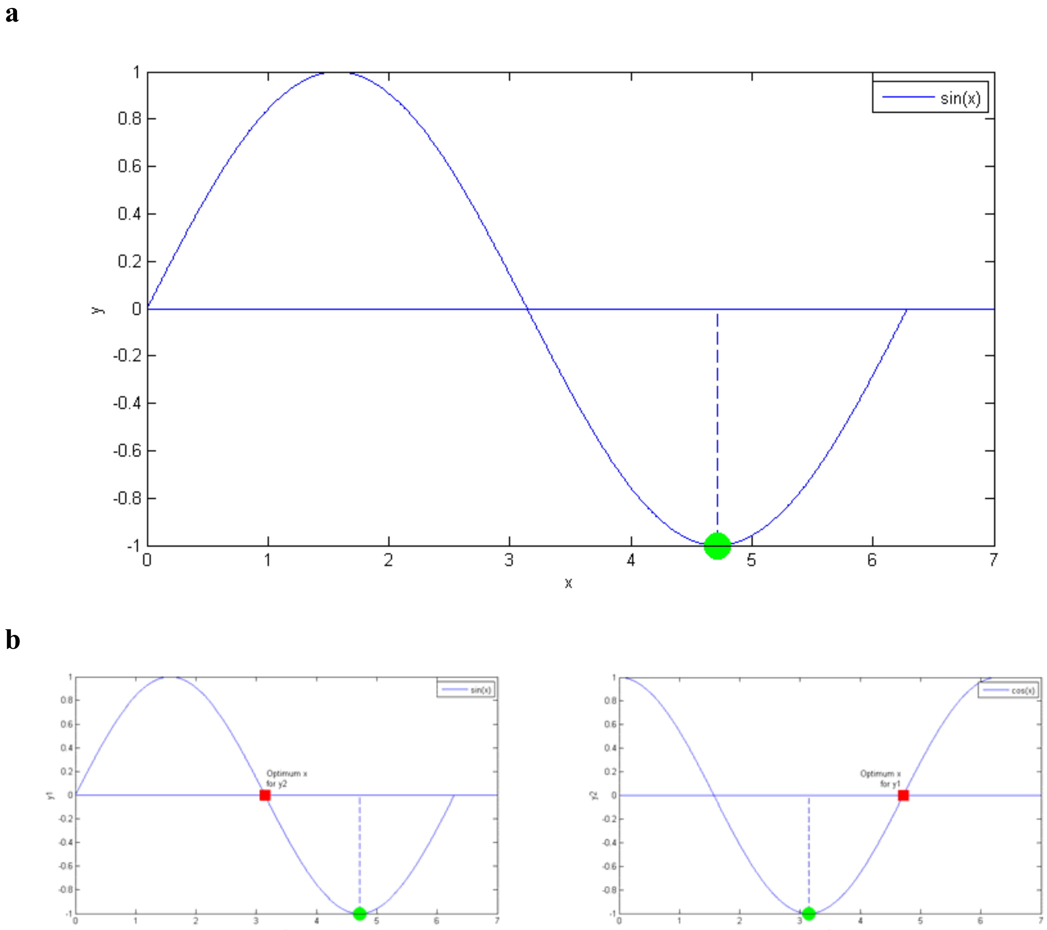


Fig. 5 (a) Optimum (dot at x=4.71)) of the simple single objective function y=sin(x). (b) Example of two objective functions in a simple multi-objective optimization problem.

Note: For single-objective optimization problems, there may exist several optimal solutions in the decision space $\mathbf{X}$. These solutions can be either **local** or **global**, where a local minimum represents a minimum in a (small) subspace of the considered decision space $\mathbf{X}$, and a global minimum represents the overall minimum in the whole considered decision space $\mathbf{X}$.

Multi-objective optimization, which can be considered as a generalization of single-objective optimization, deals with vector-valued objective functions $\mathbf{y} = \mathbf{f}(\mathbf{x})$. The definition of optimality in this case is non-trivial, because of the indefiniteness of the relation among the multiple objective functions. Hence, comparing two solutions of the vector valued objective function, e.g. $\mathbf{y}^1$ and $\mathbf{y}^2$ in the points $\mathbf{x}^1$ and $\mathbf{x}^2$, is not straight-forward. For example, in the simple case where we have two objective functions, $\mathbf{y} = (y_1, y_2)$, both to be minimized, then the first objective function y_1 may be **lower** in $\mathbf{x}^1$ than in $\mathbf{x}^2$, while the second objective function y_2 is **greater** in $\mathbf{x}^1$ than in $\mathbf{x}^2$. Alternatively, in another point $\mathbf{x}^3$, y_2 may be lower than in another point $\mathbf{x}^4$, while y_1 is not.

As an illustration, consider the following simple multi-objective optimization problem:

$$\min_{x} \; y_1 = \sin(x) \quad , \quad y_2 = \cos(x) \quad ; \quad x \in [0,2\pi]$$

The optimum points of each of these two objective functions separately are indicated in the figure 5B below (dots at y_1 and y_2 = -1); the values of the one function in the optimum of the other function are also indicated (dots at y_1 and y_2 = 0).

Obviously the optima of the two objective functions do not coincide, so there is no clear unique solution to the multi-objective optimization problem.

To resolve such kind of multi-objective optimization problems, consider the following definition, which is based on the concept of Pareto optimality[30] (Fig. 6a). According to this concept, an objective vector $\mathbf{y}^1$ is said to *dominate* any other objective vector $\mathbf{y}^2$ $(\mathbf{y}^1 \prec \mathbf{y}^2)$ if the following two conditions hold:

1. no component of $\mathbf{y}^1$ is greater than the corresponding component of $\mathbf{y}^2$,
2. at least one component of $\mathbf{y}^1$ is smaller than $\mathbf{y}^2$.

Accordingly, we can say that a solution $\mathbf{x}^1$ is *better* than another solution $\mathbf{x}^2$, i.e., $\mathbf{x}^1$ *dominates* $\mathbf{x}^2$ $(\mathbf{x}^1 \prec \mathbf{x}^2)$, if $\mathbf{y}^1 = \mathbf{f}(\mathbf{x}^1)$ dominates $\mathbf{y}^2 = \mathbf{f}(\mathbf{x}^2)$. For example, let decision vector $\mathbf{x}^1 = -1$ and $\mathbf{x}^2 = 1$, and their corresponding objective vectors are $\mathbf{y}^1 = (1,1)$ and $\mathbf{y}^2 = (9,1)$, i.e. in that case $\mathbf{y}^1$ dominates $\mathbf{y}^2$. Additionally, a solution vector $\mathbf{x}^u \in \mathbf{X}$ is said to be *Pareto optimal* if there exists no $\mathbf{x}^v \in \mathbf{X}$ for which $f(\mathbf{x}^v)$ dominates $f(\mathbf{x}^u)$. The set of (Pareto) optimal solutions in the decision space $\mathbf{X}$ is in general denoted as the *Pareto optimal set* $\mathbf{X}^* \subseteq \mathbf{X}$, and we denote its image in objective space as *Pareto front* $\mathbf{Y}^* = f(\mathbf{X}^*) \subseteq \mathbf{Y}$. This concept of Pareto optimality is illustrated in Fig. 6a below.

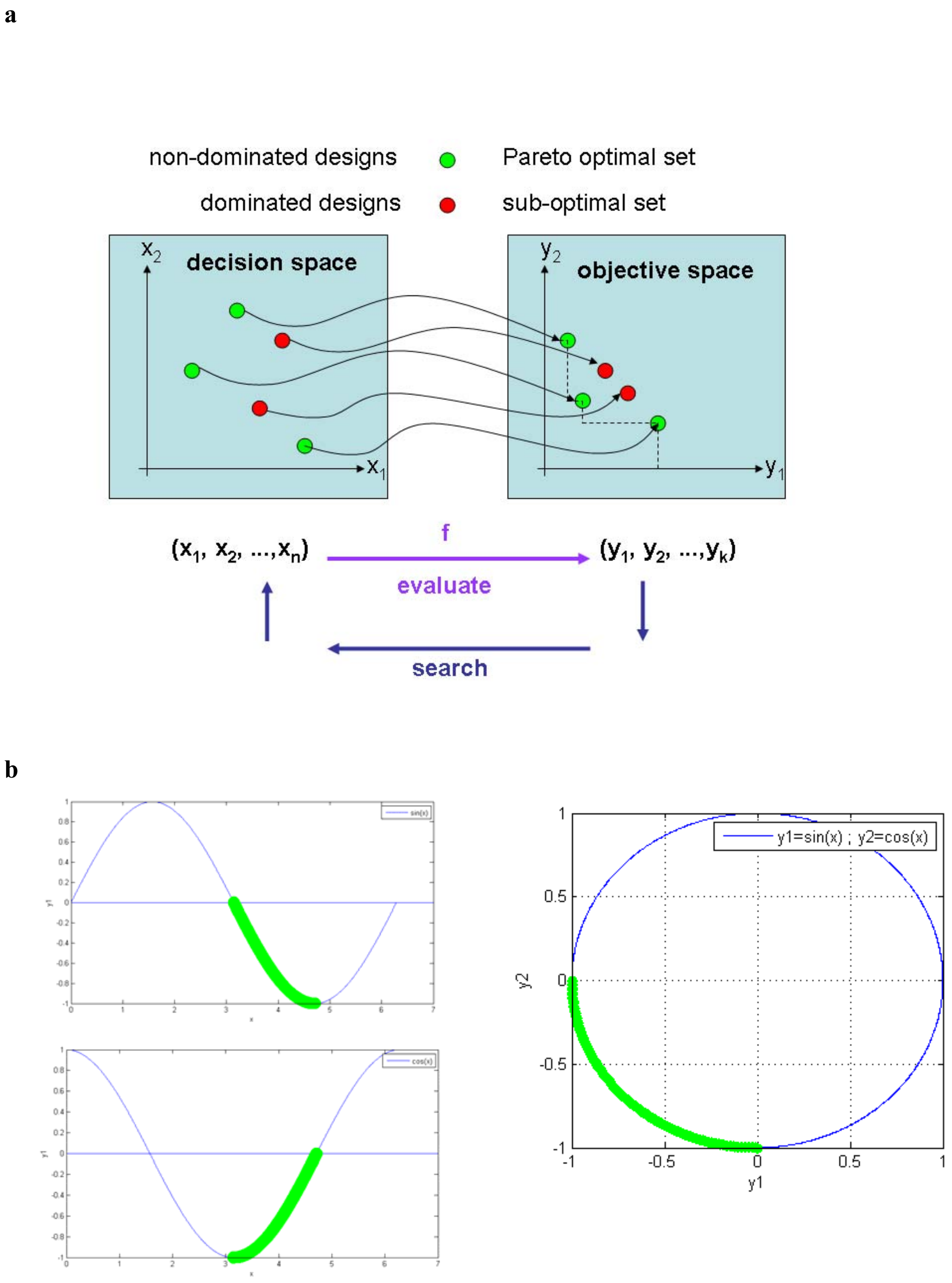


Fig. 6 (a) Illustration of the solution of a general multi-objective optimization problem. (b) Example of the solution of the simple multi-objective optimization problem.

The simple multi-objective optimization problem presented above is now handled according to the concept of Pareto optimality. The solution then consists of the Pareto front for y_1, y_2 as indicated by the thick lines in the figure 6B below ($x \in [\pi, \frac{3\pi}{2}]$).

In summary, the solution of an MOO problem is adequately defined by the *Pareto optimal* definition, which can be considered as a multi-dimensional generalization of a single objective optimum.

There exist many different algorithms to find the *Pareto optimal set*. However, in accordance with the No Free Lunch Theorems[33], there is no best algorithm for all classes of optimization problems. Therefore different methods, which give different performance for different classes of optimization problems, will be briefly presented and used in this study. The focus will be on the so-called evolutionary algorithms. These algorithms have developed strongly in the last decade, and have shown to be very effective for multi-objective optimization problems. Also, various commercial implementations of evolutionary algorithms for multi-objective optimization problems are currently available, e.g. in Matlab[25].

B Evolutionary Algorithms and Multi-Objective Optimization (MOO)

Engineering optimization problems involve a number of characteristics due to which these problems are difficult to be solved by classical numerical optimization algorithms[17]. Some of these characteristics are:

- existence of multiple conflicting objectives,
- existence of multiple optimums (local and global),
- existence of non-linear constraints,
- non-smooth or even non-continuous functions,
- stochastic and uncertainties in functions describing the optimization problem.

The term *Evolutionary Algorithm* stands for a class of stochastic optimization methods that mimic nature’s evolution. Therefore these algorithms borrow some terminology from natural evolution. For instance, we call solution vectors *individuals,* a set of individuals is called *population*, and the objective function is called *fitness*. Roughly speaking, a general stochastic search algorithm consists of three parts: i) a working memory that contains the currently considered solution candidates, ii) a selection module, and iii) a variation module as depicted in Fig. 7[15] while *mating selection* mimics the competition for reproduction and *environmental selection* mimics the “survival of the fittest” among living beings, the other principle, variation, imitates the natural capability of creating ”new” and “improved” living beings by means of so-called *recombination* and *mutation* operators[15].

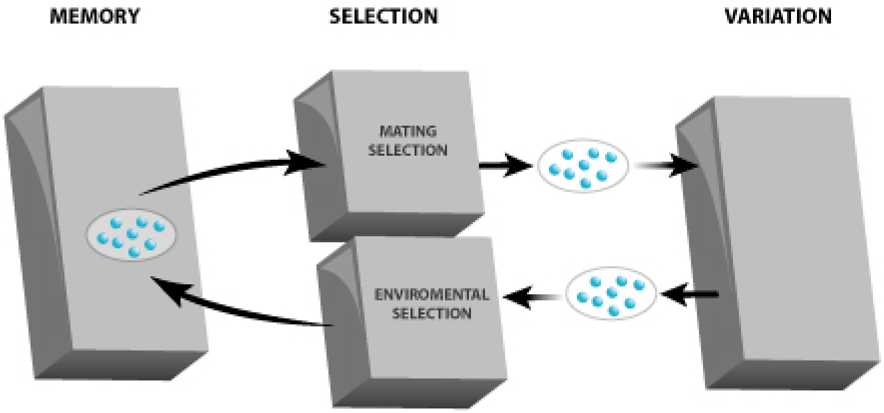


Fig. 7 Components of a general stochastic search algorithm[15]

An *evolutionary algorithm* is characterized by three features[15]:

1. a set of solution candidates is maintained (memory, Fig. 7),
2. a mating selection process is performed on this set (selection, Fig. 7), and
3. several solutions may be combined in terms of recombination to generate new solutions (variation, Fig. 7).

Several evolutionary algorithm methodologies such as *genetic algorithms*, *evolutionary programming*, and *evolution strategies* have been proposed[15]. In this chapter, we look mainly at genetic algorithms (GAs).

In general, the evolutionary algorithm process is as follows. Say, at the first *generation* we have a population consisting of randomly created individuals. All these individuals have values for their objective functions (or *fitness*). The evolutionary algorithm performs a *selection* process to select the most appropriate individuals (individuals that have the best fitness values). All the best individuals found are kept in memory for the second generation (see Fig. 7). The question now is how to produce new individuals that will be better than the best individuals found in the first generation. Here, the natural evolution principles for variation are adopted in the algorithm. First, we consider *recombination*. Recombination is the process where from two selected individuals, called *parents*, certain parts are recombined to form new individuals, called *children*. Hence, a new pair of individuals is created by a recombination function ψ_{recomb} according to: $(\mathbf{x}^{new1}, \mathbf{x}^{new2}) = \psi_{recomb}(\mathbf{x}^1, \mathbf{x}^2)$ The second process is called *mutation*. Mutation is applied in order to enforce genetic diversity from one generation to the next. In many evolutionary algorithms mutation is applied with a certain probability to the individuals (children) created by the recombination process. The mutation process creates one new individual by just changing small parts of a selected individual. Hence, a new individual is created by a mutation function ψ_{mut} according to: $\mathbf{x}^{new} = \psi_{mut}(\mathbf{x}\ \ )$ These two steps of selection and variation are iteratively repeated until a converged optimum solution is found, or until a predefined maximum number of iterations (or generations) have been achieved.

There exist many multi-objective evolutionary algorithms (MOEAs) which have recently been proposed in the literature, some of which are considered here more closely: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [18], the ε-dominated Multi-Objective Evolutionary Algorithms, ε-MOEA[19], and the ε- dominated Non-dominated Sorting Genetic Algorithm ε-NSGA[32]. These algorithms will be investigated in more detail because these algorithms have the most promising properties for aircraft design optimization.

In both NSGA-II and ε-MOEA algorithms, similar recombination and mutation operators are applied. They use *simulated binary crossover*[34] for recombination and *polynomial mutation*.

With simulated binary crossover (SBX) the children $(\mathbf{x}^{(1,t+1)}, \mathbf{x}^{(2,t+1)}) = \psi_{recomb}(\mathbf{x}^{(1,t)}, \mathbf{x}^{(2,t)})$ of two parents of generation *t* are computed with the following equations:

$$\begin{aligned} x_i^{(1,t+1)} &= 0.5[(x_i^{(1,t)} + x_i^{(2,t)}) - \beta_{qi}(x_i^{(2,t)} - x_i^{(1,t)})] \\ x_i^{(2,t+1)} &= 0.5[(x_i^{(1,t)} + x_i^{(2,t)}) + \beta_{qi}(x_i^{(2,t)} - x_i^{(1,t)})] \end{aligned} \tag{15}$$

and

$$\gamma_i = \begin{cases} 1 + 2\dfrac{x_i^{(1,t)} - x_i^L}{x_i^{(2,t)} - x_i^{(1,t)}} & \text{for child 1} \\ 1 + 2\dfrac{x_i^U - x_i^{(2,t)}}{x_i^{(2,t)} - x_i^{(1,t)}} & \text{for child 2} \end{cases} \tag{16}$$

Where x_i^U and x_i^L are the lower and upper bounds on the decision variables x.

The non-negative user defined parameter, *distribution index for crossover* η_c controls the shape of the probability distribution of the spread of the children. A large value of η_c gives a higher probability for creating near-parent solutions and a small value of η_c allows distant solutions to be selected as offspring.

$$\beta_{qi} = \begin{cases} (u_i\alpha_i)^{\frac{1}{\eta_c+1}} & \text{if } u_i \le \dfrac{1}{\alpha_i} \\ \left(\dfrac{1}{2-u_i\alpha_i)}\right)^{\frac{1}{\eta_c+1}} & \text{otherwise} \end{cases} \tag{17}$$

With u_i a random number between 0 and 1, and

$$\alpha_i = 2 - \gamma_i^{-(\eta_c+1)} \tag{18}$$

From Equation (17) – (18), one can see that the first child will lie between lower bound and midpoint between first and second parents, while the second child will lie between midpoint between first and second parents and upper bound. This implies that if two parents are far away, a usual condition for the initial population, almost any value of offspring can be achieved. When the solutions tend to converge, the parents are close to each other, distant solutions are not allowed, thereby focusing the search to a narrow region. Thus, SBX helps in exploring the search space at the initial generation, while it exploits the acquired knowledge at later stages.

In polynomial mutation[15] the following steps are involved in mutating a solution $x_i^{(1,t)}$ to obtain the offspring $x_i^{(1,t+1)}$. Let u_i be a random number between 0 and 1.

$$\overline{\delta_i} = \begin{cases} (2u_i)^{\frac{1}{\eta_m+1}} - 1 & \text{if } u_i < 0.5 \\ 1 - [2(1-u_i)]^{\frac{1}{\eta_m+1}} & \text{otherwise} \end{cases} \tag{19}$$

$$x_i^{(1,t+1)} = x_i^{(1,t)} + (x_i^U - x_i^L)\overline{\delta_i} \tag{20}$$

The shape of the probability distribution of the distance between the mutation and its original is directly controlled by the user defined parameter, *distribution index for mutation* η_m. A large value of η_m gives a higher probability for creating near-parent solutions and a small value of η_m allows distant solutions to be selected as offspring.

C NSGA-II

The NSGA-II procedure[18] for finding multiple Pareto optimal solutions for multi-objective optimization problems has the following three main features:

1. It is based on Pareto dominance sorting,
2. Elitism by preservation of best individuals,
3. Diversity preserving mechanism by crowding distance sorting (distance here represents the Euclidian distance between an individual and its closest neighbor).

These processes are further explained below.

Initially, NSGA-II randomly creates a population containing N individuals. This population, $P(0)$ is sorted according to their rank values to get $P(1)$ .The sorting is based on the order of dominance of the individuals. An individual that is not dominated by any other individuals gets a Pareto rank 1. Subsequently the individuals with rank 2 are the ones that are dominated only by the individuals with rank 1, etc. After sorting, the offspring population $Q(1)$ (i.e. children of P(1)), is created by selecting parents from population $P(1)$ and applying the genetic recombination and mutation operators[34, 35], as described above. Thereafter, the two populations $(P(1), Q(1))$ are combined together to form the new population R_1 of size $2N$. Because this combined population R_1 is used, the best individuals of both the parents and children can be preserved, yielding a so-called *elitism* mechanism.

Then again, a non-dominated sorting is applied to the entire population R_1. The new population is then filled by individuals of subsequent ranks (i.e. non-dominated fronts F_1, F_2, etc.; see Fig. 8) starting with rank 1, followed by rank 2, etc.

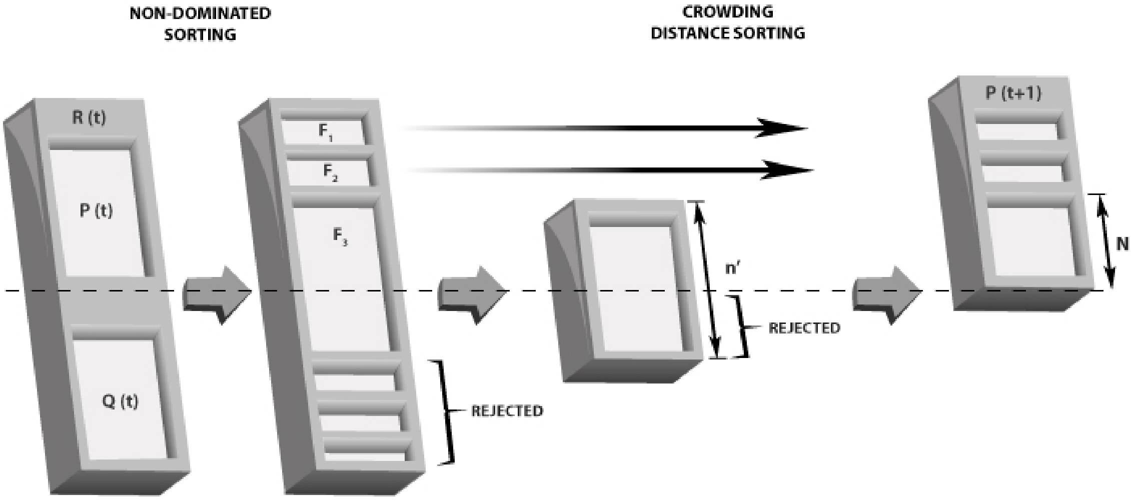


Fig. 8 The NSGA-II procedure[18]*.*

Since the overall population size of R_1 is $2N$, not all fronts can be accommodated in the N slots (i.e. individuals) available in the new population. All fronts that are not accommodated at all are simply deleted. The front that can be accommodated only partly is treated separately. In Fig. 8 we see that not all individuals from front F_3 are included in the next generation. NSGA-II deals with this as follows. Per objective function, all individuals in the front F_3 are sorted according to their objective function values in ascending order. The minimum and maximum are selected and given infinity distance. These minimum and maximum will be put first on the list. This is to ensure that boundary points are always selected. Secondly, the distances for the rest of individuals are calculated by means of *cuboids*, formed by using the nearest neighbors of an individual as the vertices. After the rest of individuals' distances are calculated, they are sorted in descending order to form the newly ordered front F_3. In Fig. 8, this is illustrated as the small block right next to F_3. The individuals with smallest distance are rejected and the rest are included into the next generation $P(2)$. This process is called *crowding distance sorting,* and it is intended to maintain a good distribution of the individuals over the full Pareto front (i.e. to avoid convergence into 1 or a few Pareto optimal points).

After all N individuals in $P(2)$ are obtained, the offspring population $Q(2)$ is created. These populations are again combined to form R_2 containing $2N$ individuals. Then the same procedure is applied for every generation until a maximum number of generations, specified by the user is reached.

D - ε-MOEA

ε-MOEA is an evolutionary algorithm based on the ε-dominance concept introduced in[19]. The properties of this algorithm are:

- It is a *steady-state* MOEA. This means that it maintains *spread* (i.e. distribution of individuals over the Pareto front) while attempting to *converge* to the true Pareto optimal,
- It emphasizes non-dominated solutions by using usual dominance concept,
- It maintains the diversity in the archive by allowing only one solution to be present in each pre-assigned hyper-box of size ε on the Pareto-optimal front (ε-dominance),
- It uses an elitism principle by maintaining all the best individuals in an archive throughout the computation.

In ε-MOEA, the objective space is divided into a number of grid cells (or hyper-boxes), as illustrated on the left of Fig. 9. The diversity of the population is maintained by ensuring that there is only one solution in a grid cell (black dots). There are two co-evolving populations: an evolutionary algorithm population, $P(t)$, and an archive population, $E(t)$, as shown on the right of Fig. 9, where, t is the iteration counter and the archive population is intended to hold the best individuals that have been found until iteration t. ε-MOEA begins with a (randomly generated) initial population $P(0)$. The archive population $E(0)$ is assigned with the ε-non-dominated individuals of $P(0)$. The ε-dominance concept is shown on the top of Fig. 9 and also in Fig. 10.

Thereafter, two solutions, one from each $P(0)$ and $E(0)$ are chosen for mating. To choose a solution from $P(0)$, two population members from $P(0)$ are picked up at random and a domination check (in the "usual" sense, shown as dots on the left of Fig. 9 for minimization of objectives) is made. If one solution dominates the other, the former is chosen. Otherwise, it indicates that these two solutions are non-dominated to each other and simply one of them is chosen at random. Let us denote the chosen solution by p. To choose a solution e from $E(0)$, several strategies involving a certain relationship with the chosen p can be made[19]. In[19], for example, they randomly pick a solution from $E(0)$. Then solution p and e are mated to create λ offspring solutions, c_i $(i = 1,2,\ldots,\lambda)$. In the present study $\lambda = 1$ is always used, as also advised in[18].

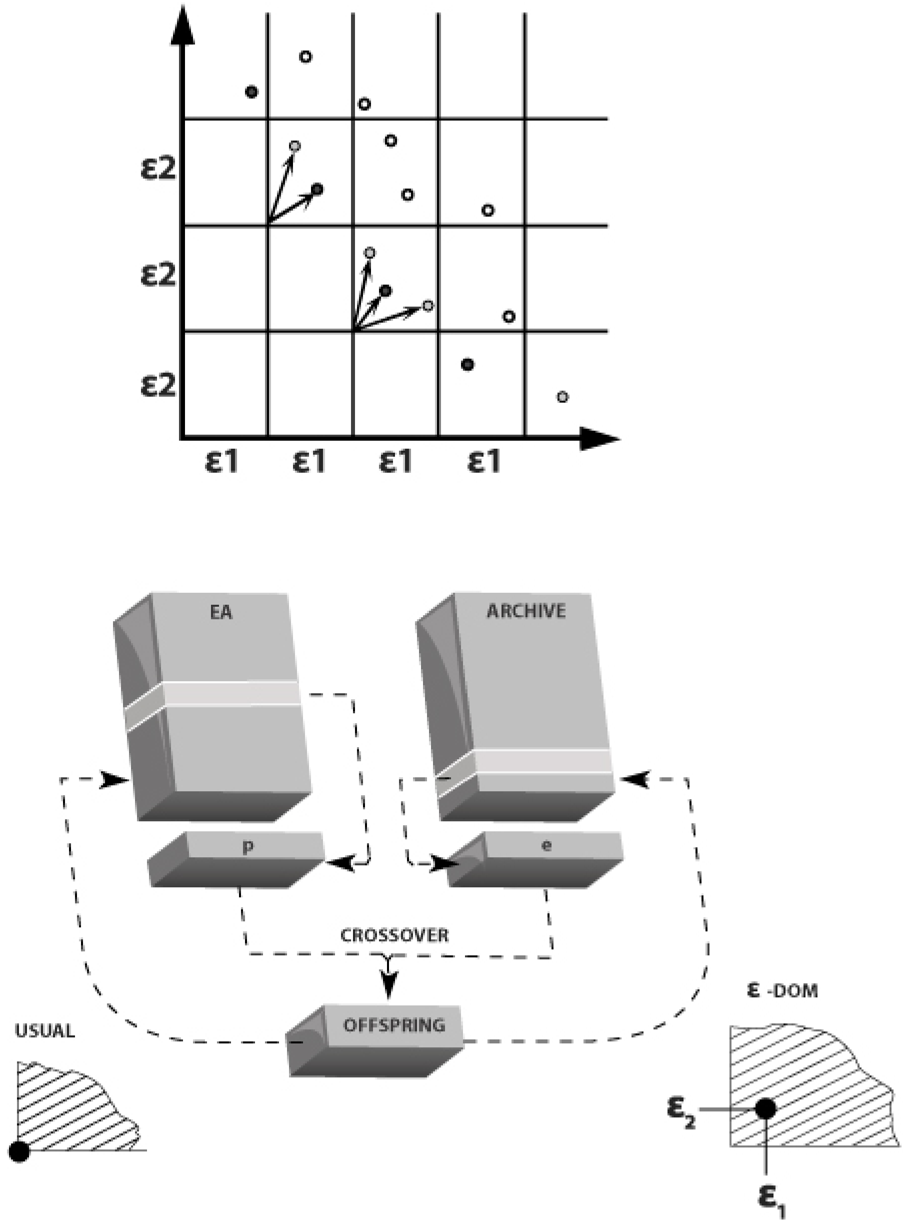


Fig. 9: ε-MOEA objective space (left) and evolution procedure[19] are illustrated

The offspring solution, c_i, is compared to the archive members by ε-non-dominated checking, based on ε-dominance concept[19]. The solutions are first compared on the basis of the Pareto dominance of the grid cells that they reside in. Subsequently, multiple solutions within one grid cell are compared on the basis of the usual Pareto dominance of these individuals. If the offspring dominates one or more archive members, the offspring replaces one of them (chosen at random). Every solution in the archive is assigned an identification array, $\mathbf{B}$, which indicates the fitness score of a grid cell and which size equals to the total number of objectives as follows:

$$B_i(\mathbf{f}) = \lfloor (f_i - f_i^{\min}) / \varepsilon_i \rfloor \tag{21}$$

where, $\lfloor \ \rfloor$ represents the truncation operation to a whole number, $f_i^{\min}$ is the minimum possible value of the *i*-th objective of all archive members and ε_i is the allowable tolerance in the *i*-th objective. This ε_i value is the same as the ε used in the ε-dominance definition[19]. The identification arrays are calculated for each archive member, a, and for the offspring c_i. If the $\mathbf{B_a}$ of any archive member a dominates that offspring c_i, then c_i is not accepted since it means that this offspring c_i is ε-dominated by archive. On the other hand, if $\mathbf{B_{c_i}}$ of the offspring dominates $\mathbf{B_a}$ of any archive member a, the archive member is replaced by the offspring. This is why individuals 3 and 4 in Fig. 10 are not accepted as an archive member. Although in the usual Pareto dominance sense they are non-dominated solutions, they are dominated solutions in ε-dominance sense since their identification arrays are dominated by their neighbors. The grid cell of individuals 1 and 2 dominates the grid cell of individual 3.

If both the offspring and the archive are ε-non-dominated which means neither of the above two cases occur, another two operations are performed. If the offspring belongs to an unoccupied grid cell (different identification array), then offspring c_i is added into the archive. If it is in the same grid cell as an archive member (having same identification array), a check for the usual non-domination is conducted. It might appear that the offspring and the archive member are non-dominated like individuals 1 and 2 in Fig. 10. If this is the case, then the one which is closer to the $\mathbf{B}$ vector in terms of the Euclidean distance is chosen (in this case individual 2) and the other (individual 1) is not kept in the archive.

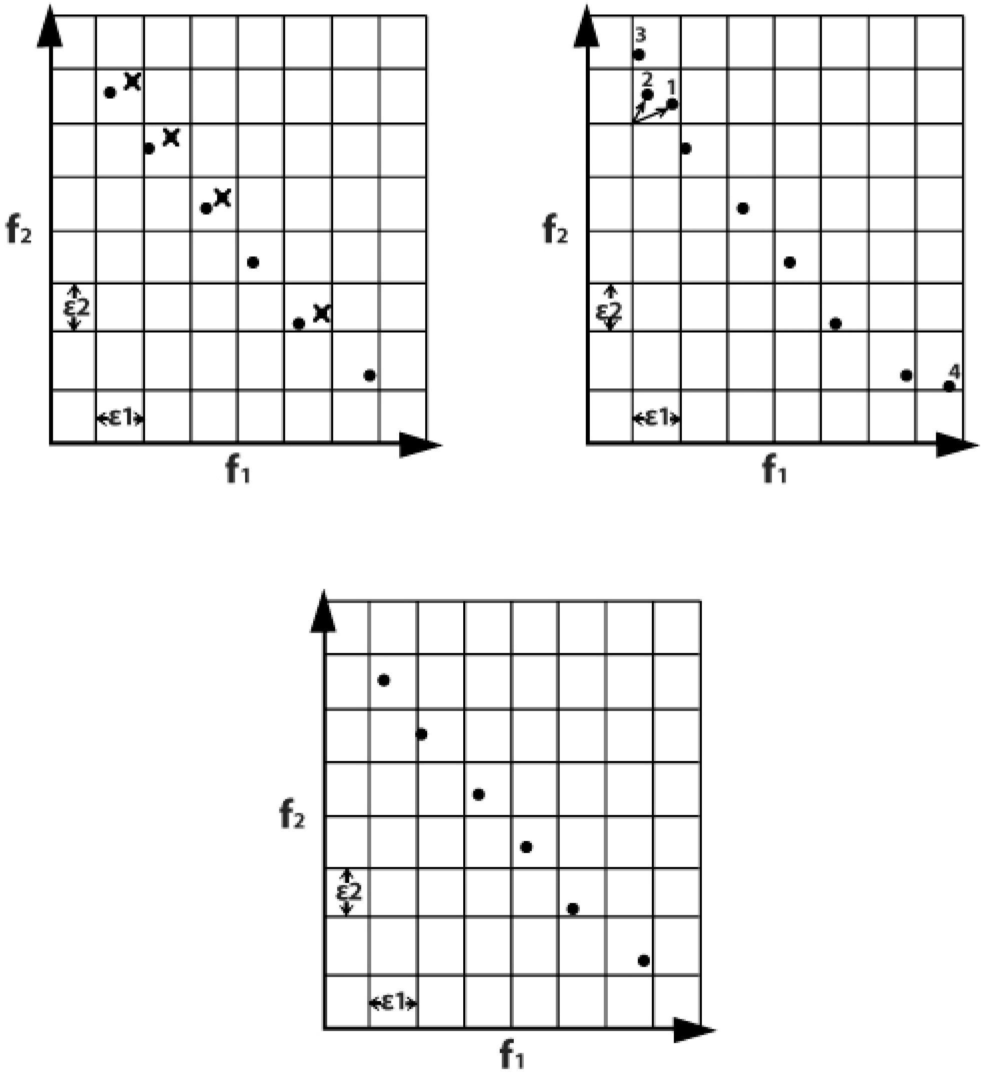


Fig. 10 Illustration of ε-dominance concept

The decision whether an offspring will replace any population member P can be made using different strategies. One of these is to compare each offspring with all population members. If the offspring dominates one or more population members, the offspring replaces one of them (chosen at random). On the other hand, if any population member dominates the offspring, it is not accepted. If both the offspring and the population members are non-dominated, the offspring replaces a randomly chosen population member so that the evolutionary algorithm population size remains unchanged.

The above procedure is continued for a specified number of iterations and the final archive members are considered as the obtained ε-non-dominated solutions.

F Improvements of ε-MOEA: ε-NSGA

There are some weaknesses of ε-MOEA:

1. the absence of *extreme* solutions, i.e., the solutions with high and low objective values, on the Pareto-optimal front;
2. for very many and expensive function evaluations, it turns out to be not efficient.

The absence of extreme solutions occurs as a consequence of the hyper-boxes. Although this issue is not always of critical importance, because the extreme solutions region is not always the most interesting design region, this issue can be alleviated by decreasing the epsilon values.

Since ε-MOEA calculates the objective function values for each individual separately, it means that we can not take the advantage of vectorized calculation. Especially if objective function calculation is expensive this is not efficient. Therefore we combine ε-MOEA and NSGAII (we name it ε-NSGAII) as shown in Fig. 11. Although such a combination has been proposed in[32] we use different method in combining these two algorithms. In the proposed algorithm, a dynamic population size is used in comparison to the archive size. We found that this method is good to explore additional regions of the search space in the early generations. But in the later stage, as the archive size increases the population size is getting bigger and bigger until the archive size is stable. At this point, we found that for expensive function calculations, this method is inefficient.

According to the reasons above, we use fixed population size so that in the later stage the algorithm can still do the calculations efficiently. In order to explore additional regions of the search space in the early generation, we randomly generate new individuals as long as the archive size does not exceed half the population size. We found that this archive size limit of “half the population size” performs best for a population of up to 100 individuals.

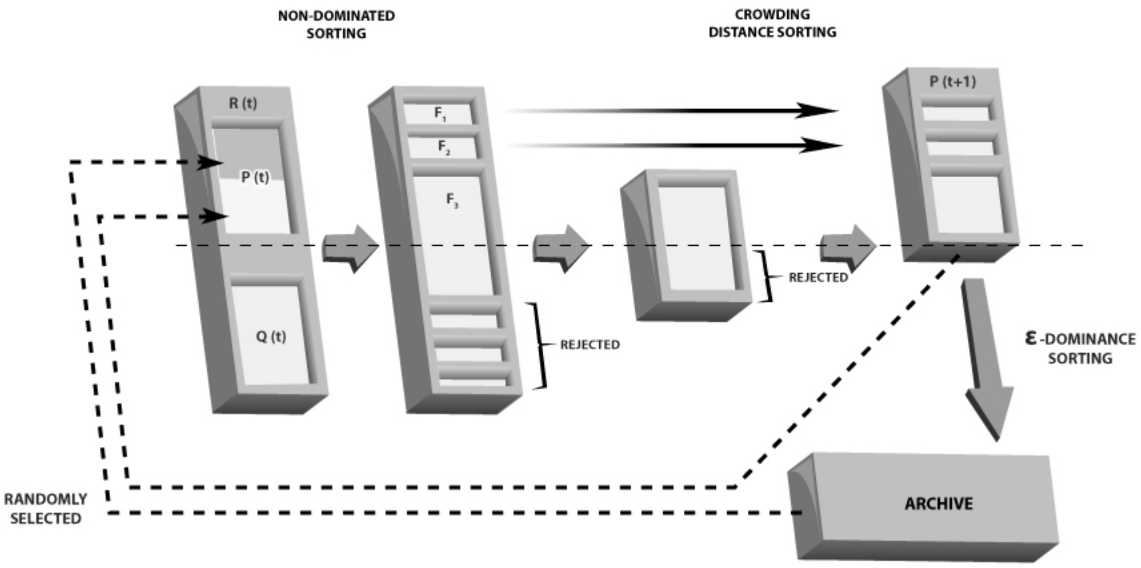


Fig. 11 Illustration of the ε-NSGAII algorithm

The ε-NSGAII (Fig. 11) uses NSGAII algorithm for generating new offspring individuals. After it finds new better individuals, ε-dominance sorting is applied and the results are stored in the archive. New individuals to evolve in the next generation are randomly selected from both the archive and the new better individuals from the current generation. Here, we use 50% population from the current generation and another 50% population is taken from the archive. If the archive size is less than 50% population, it generates new individuals randomly.

IV Case study: Multi-objective optimization of transonic wing design

A Introduction of case study

This section applies the meta-models and optimization algorithms that were described in the previous sections, in an example aircraft wing optimization study.

The case study considered in this chapter describes an investigation of aircraft range and fuel efficiency. Fuel efficiency here represents the distance flown per unit of fuel per unit of payload. The investigation concerns the optimization of transonic aircraft wings in the preliminary design phase. The design analyses in this investigation make use of a multidisciplinary design analysis (MDA) system that evaluates the aircraft characteristics as a function of a set of design parameters[36]. The evaluations comprise, among others, wing structural sizing and optimization using finite element method (FEM) analyses, and cruise lift over drag performance using computational fluid dynamics (CFD) analyses. The design parameters include geometric wing planform parameters such as span, chord, sweep, as well as "aircraft operational parameters" such as maximum take-off weight (MTOW) and cruise altitude. Given the design parameters inputs, the wing MDA system predicts the corresponding aircraft characteristics in terms of, among others, weight breakdown information, maximum range and fuel consumption. More details of this MDA system are given in the chapter on wing multidisciplinary analysis.

In order to effectively handle the different (and possibly conflicting) design objectives, multi-objective Pareto front[30] optimization algorithms as described above are used in the presented aircraft wing design investigation. The multi-objective Pareto front results directly provide the design information on which further trade-off considerations of the different objectives for the wing design can be based.

To limit the number of computationally expensive evaluations with the MDA system, the multi-objective optimization iterations have been de-coupled from the MDA evaluations, according to the meta-modeling approach described above (see also[37]). The meta-modeling approach allows for computationally efficient exploration of the aircraft characteristics in a pre-defined design domain. The different meta-modeling methods, such as polynomial regression, kriging models and neural networks, are used and their predictive accuracy is carefully checked and compared in order to achieve the best representation. Obviously, the results of the optimization depend on the accuracy of the meta-models used, and therefore also require careful assessment and validation, as is shown in the present aircraft design optimization study.

B Aircraft multi-disciplinary design analysis

The MDA system described in the previous chapter is used in an aircraft wing design optimization study. Aircraft designs are pursued that have optimal performance for both range and fuel efficiency.

From the many results that come out of the MDA simulations, different variables can be selected as relevant objective or constraint functions in aircraft design optimization studies. In the present wing design optimization study we look for optimal overall range and fuel efficiency, and hence we take into account the Breguet range and the total fuel consumption as the aircraft wing optimization objectives, which are both computed with the MDA system The Breguet range (R_B) represents the actual distance travelled, taking into account the engine fuel efficiency during cruise (c_{fs}), the aerodynamic lift-over-drag performance $L_{oD\text{-}cruise}$, and the actual amount of fuel consumed (W_{fc}) as follows:

$$R_B = \frac{v_{cruise}}{c_{fs}} \times L_{oD-cruise} \times \ln\left(\frac{W_{MTO}}{W_{MTO} - W_{fc}}\right). \tag{22}$$

It should be noted here, that the fuel weight (W_{fc}) depends on the aircraft structural weight, which is computed in the MDA through a structural optimization for the given aircraft configuration. The aircraft fuel efficiency η_f can be evaluated as a combination of range and actual fuel consumption, and is calculated by:

$$\eta_f = \frac{R_B}{(\frac{W_{fc}}{n_{pax}})}, \tag{23}$$

and is expressed in [km/(l/person)]. These values can be easily compared to other fuel efficiency numbers as for example published for cars ($\eta_f \sim 14$ for single person driving a middle class car).

From the many possible design parameters that are used in the MDA system, we selected four illustrative ones as the independent variables for the wing optimization: wing semi-span, outer wing leading-edge sweep angle, wing chords, and aircraft MTOW (Fig. 12). The three wing chords (at root, crank and tip) are reduced to a single parameter, the wing chord scale factor, which linearly scales all three chords equally. All other design parameters of the MDA system are equal to their values for the reference aircraft and remain unchanged in this study.

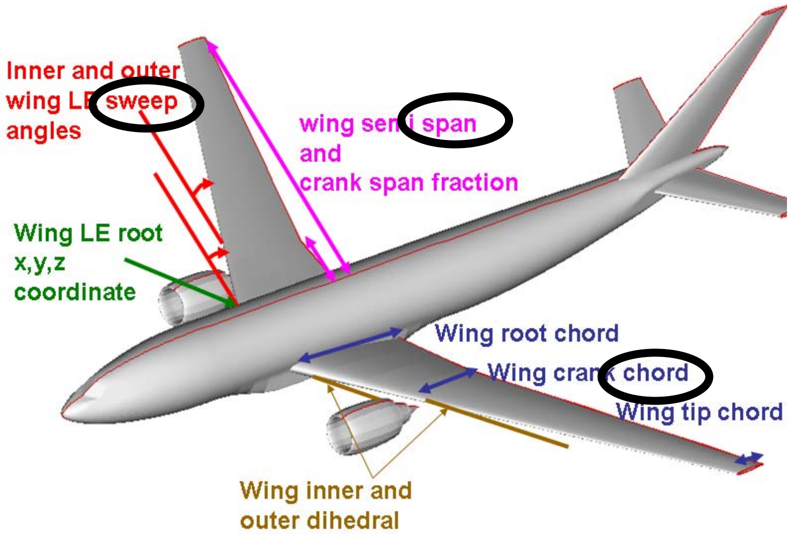

Inner and outer wing LE sweep angles
wing semi span and crank span fraction
Wing LE root x,y,z coordinate
Wing root chord
Wing crank chord
Wing tip chord
Wing inner and outer dihedral

Fig. 12 Aircraft wing design parameters.

The resulting multi-objective optimization problem for aircraft range and fuel efficiency can be formulated as follows:

$$\max_{(s_p, s_w, c_h, W_{MTO})} (R_{Bcorr}, \eta_f). \tag{24}$$

C Meta-models

In order to create the meta-models, first a suitable sample of the aircraft behavior in the considered design domain is pursued. This is achieved by a limited number of evaluations with the MDA simulation system in certain selected design points. These design points are defined according to a sequence of fractional factorial (i.e., fractions of full-factorial) sets of samples of the four dimensional design space (i.e., parameter space of the design parameters: wing semi-span, outer wing sweep angle, wing chord, and aircraft MTOW). The semi-span is varied between 29 m and 32 m. The outer wing sweep angle is varied between 21 deg and 39 deg. The wing chords at 3 stations (wing root, crank and tip) are equally varied by one single chord scale factor, which is varied between 1.000 and 1.075. MTOW is varied between 150000 and 280000 kg.

In total, 99 design points are created in this parameter space and are evaluated with the full MDA simulation system, yielding (among many other data available in the integrated design model) the values for range and fuel consumption in these design points. As a quick preliminary design assessment, these range and fuel values are ordered according to a basic Pareto ranking procedure[30], as described in section 4.3, in order to obtain a first indication of the interesting design regions. In this ranking procedure, the best (or non-dominated) design points, i.e. those points having the best values for range and fuel consumption, are assigned Pareto rank 1, the set of second best points are assigned Pareto rank 2, and so forth until all design points have been assigned a rank value.

The resulting rank values for these 99 design points, and their distribution in the objective space and their parameter values are given in Fig. 13 below.

The resulting data set with the values of the design parameters and of the range and fuel objectives in these 99 design points is then used to create the meta-models. The meta-models shall approximate as accurate as possible the objectives in each point of the parameter space.

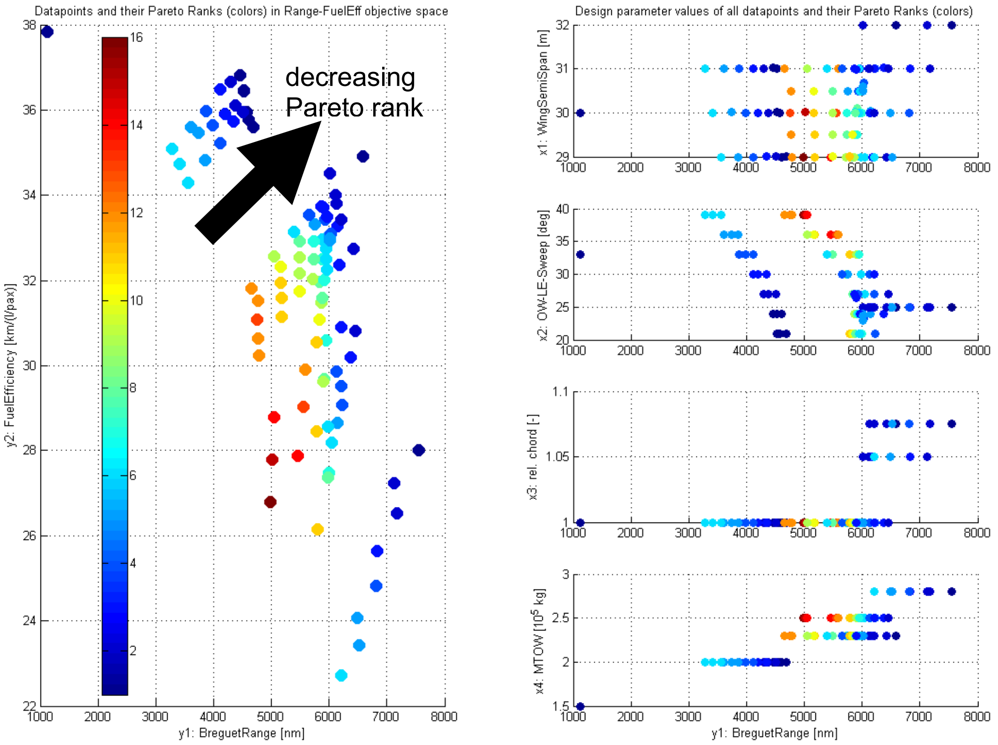


Fig. 13 The range and fuel results in the 99 design points in objective space (left) and in parameter space (right), coloured by their Pareto rank (the rank 1 points have the highest range and highest fuel efficiency values).

A number of different polynomial functions (polyn in Table 1 and Table 2), kriging models (kriging-xy in Table 1 and Table 2), neural networks (ann in Table 1 and Table 2) and radial basis functions (rbf in Table 1 and Table 2) are applied[38], and the best fit functions among these are determined. These best fit functions are found through various cross-validation assessments on the data set, such that these functions' predictions of the design objectives (range, fuel efficiency) have the smallest residuals. Four different cross-validation assessments are performed by selecting different sets of validation points.

In a first cross-validation assessment the nine rank-one data points, i.e. those data points having the best (lowest) Pareto rank values for range and fuel efficiency (dark blue dots in Fig. 13), are used as validation points, and the remaining 90 for building the model. The resulting RMS values indicate that the kriging-linear-Exponential (kle) [28] and second order polynomial (poly2) fit functions provide the best fits for range and fuel-efficiency, respectively (99/9-column in Table 1 and 2). However, this assessment represents the accuracy of the fits in only a local region around the rank-one data points, for fits where only design value points have been used that yield objective of rank two or worse. In order to obtain a more global accuracy assessment we include some more validation points by adding the 11 Pareto rank-two data points to the validation set (99/20-column in Table 1 and 2). Because this validation set is rather large (20 out of 99 points), the validation fits are made on relatively small data sets (79 points), and thus will differ significantly from the "full" fits made on the complete data set (99 points)), and again the best points are actually removed from the fitting stage. Therefore we also evaluate the RMS-residuals from a leave-1-out experiment[39] of this validation set (99/1/20-column in Table 1 and 2). In this leave-1-out experiment, subsequently each point of the validation set is separated from the data set, a fit is made on the remaining 98 points, the residual in the validation point is evaluated, and the RMS of the 20 residuals is calculated. Finally, as a real global accuracy assessment, we also performed a leave-1-out experiment on the complete data set (99/1/99-column in Table 1 and 2). As an additional indication of the relative accuracy of the fits, we also include the Mean Absolute Percentage Error (MAPE) of the global leave-1-out residuals (99/1/99/%-column in Table 1 and 2).

For the different cross validation assessments we find reasonably consistent accuracies for most fit functions (Table 1 and 2). The best RMS-residual found in each assessment is marked by the green shaded cell. For the range data (Table 1), the radial basis function (rbf) fit provides the best results for the leave-1-out experiments, but very poor fit quality according to the 99/20 experiments, and is therefore not selected as best fit for range.

Based on the results of each of the 5 assessments performed, and in particular on the global accuracy as measured by the leave-1-out experiments (Table 1, columns 99/1/99 and 99/1/99/%), it is concluded that the best fit for range is found by the kriging-linear-Gauss (klg) fit function. For fuel efficiency the poly2 fit performs quite well (Table 2), but its global accuracy as measured by the leave-1-out experiment (column 99/1/99/%) is worse than for some of the kriging fits. In addition, poly2 provides a least-square regression (non-interpolating) fit on the data, whereas the kriging models provide exactly interpolating fits on the data. Because the data represents results of deterministic computer simulations, it is concluded that the best fit for fuel efficiency is found by the kriging-constant-Exponential (kce) fit function.

Table 1. Range data: Accuracies of the different fit functions (identified in left column) for the different cross-validation

	RMSE				MAPE
fit function	99/9	99/20	99/1/20	99/1/99	99/1/99/%
poly0	1824.8	1450.2	1464.0	993.2	18.5785
poly1	789.0	720.6	541.0	401.6	6.7994
poly2	739.3	509.2	460.8	234.1	3.7504
kriging-cG	1386.0	1155.3	886.3	400.3	4.2159
kriging-cE	1297.2	730.4	913.8	414.1	4.2473
kriging-cC	1025.6	722.3	814.8	367.0	3.8202
kriging-lG	608.7	519.3	301.7	138.6	1.7258
kriging-lE	567.6	418.8	465.5	210.1	2.2546
kriging-lC	600.9	440.5	411.0	186.8	2.2124
ann	1175.3	1053.7	957.3	859.6	12.8121
rbf	784.1	5130.0	205.0	99.7	1.1252

Table 2 Fuel-efficiency data: Accuracies of the different fit functions for the different cross-validation assessments

	RMSE				MAPE
fit function	99/9	99/20	99/1/20	99/1/99	99/1/99/%

poly0	4.648	4.182	3.909	3.259	8.4636
poly1	1.984	1.499	1.368	0.995	2.3676
poly2	0.722	0.544	0.258	0.264	0.6371
kriging-cG	1.746	1.422	0.830	0.421	0.6577
kriging-cE	2.435	1.289	0.947	0.430	0.3894
kriging-cC	2.103	2.251	1.198	0.576	0.7798
kriging-lG	1.590	1.358	0.939	0.443	0.4836
kriging-lE	1.692	1.378	1.187	0.539	0.4136
kriging-lC	1.778	1.404	1.305	0.607	0.6339
ann	1.886	1.393	0.672	1.179	3.7600
rbf	6.990	66.977	4.740	2.140	1.1210

Regarding both tables above, the values given are the root-mean-squares of the residuals (or prediction errors) in the validation points. The thick-bordered cell marks a poor fit quality. The double-bordered cells mark the best overall fit method.

D Design optimization

A Pareto front optimization of the aircraft's range and fuel efficiency is performed using a multi-objective genetic algorithm (based on ε-NSGA-II as described in the previous section), where the best fits for range and fuel efficiency are used as objective functions. In this optimization a population size of 99 individuals is used, where the 99 design points from the data set are used as the initial generation. The bounds of the search domain for the optimization are set to the minimum and maximum values of the design parameters of the 99 design points. In a first run 3 generations, so about 300 objective functions evaluations are performed with the genetic algorithm. The resulting population is indicated by the green circles in Fig. 14 in order to give some illustration of the convergence history of the genetic algorithm. Then this resulting population is used as the initial population for an extensive run of about 100 generations with the genetic algorithm. The total number of objective function evaluations in this extensive optimization is about 10.000, and takes about 20 seconds computational time on a standard PC (Pentium-4, 2.8 GHz). The resulting Pareto front solution (diamonds in Fig. 14) provides a set of clearly improved designs, as compared to the initial set of designs in the data set (small dots).

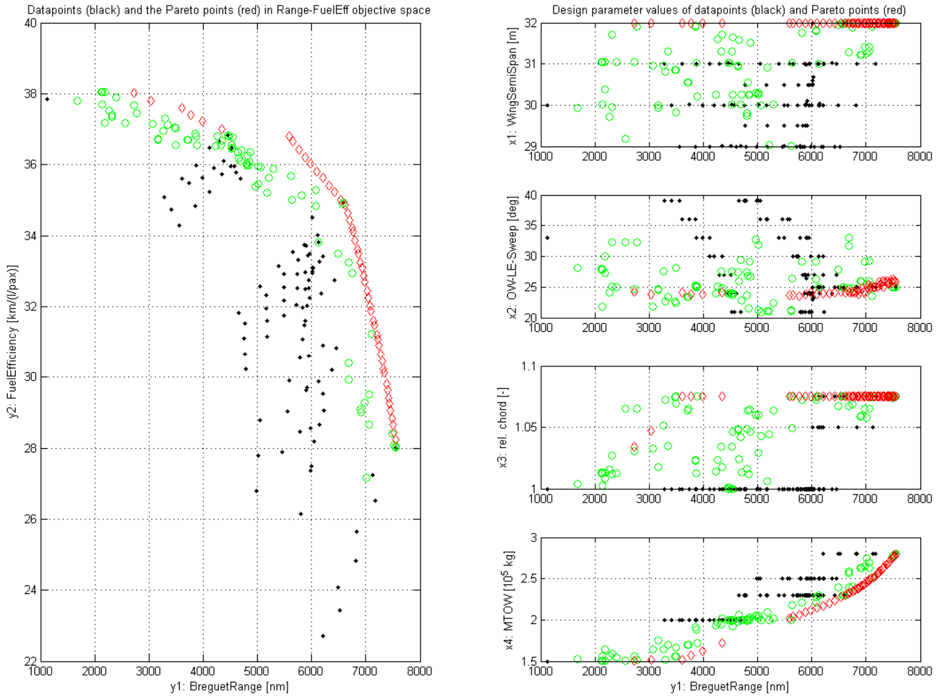
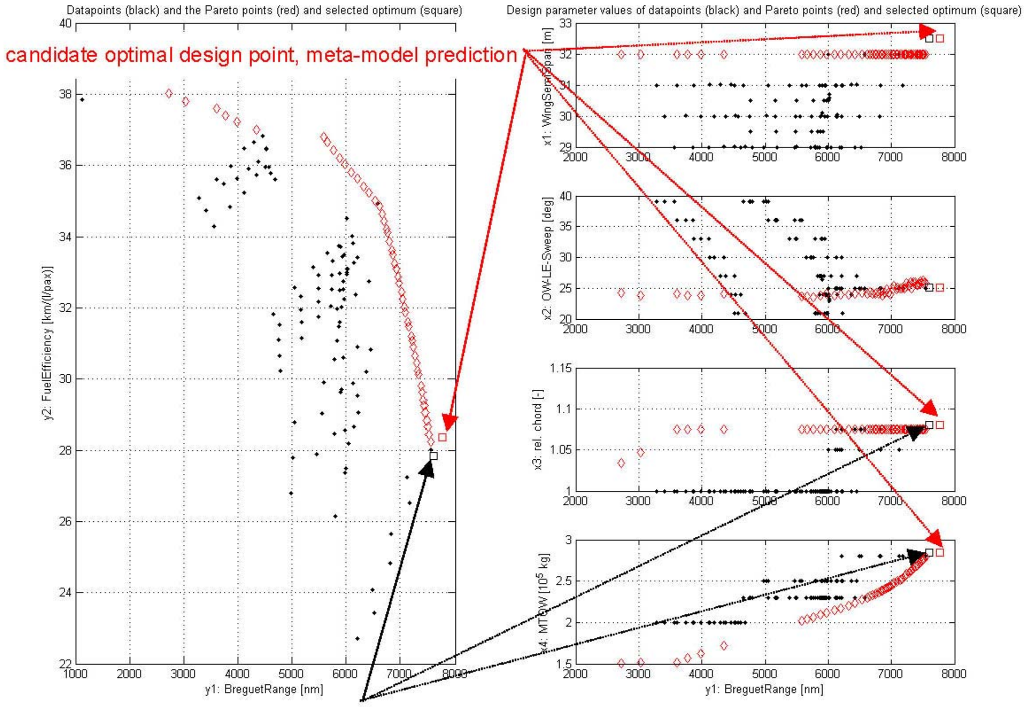


Fig. 14 Design points of data set (small dots), population after 3 generations (circles), and Pareto front after 100 additional generations (diamonds) for maximum range versus maximum fuel efficiency found with the kriging-linear-Gauss and kriging-constant-Exponential meta-models, respectively, for range and fuel efficiency. Results presented in objective space (left) and in the range- parameter sub-spaces (right) for each of the four design parameters.



candidate optimal design point, meta-model prediction

candidate optimal design point, MDO analysis

Fig. 15 Pareto front found with initial meta-model (diamonds), data set (small dots), and MDA analysis and meta-model predictions for candidate optimal design point (squares).

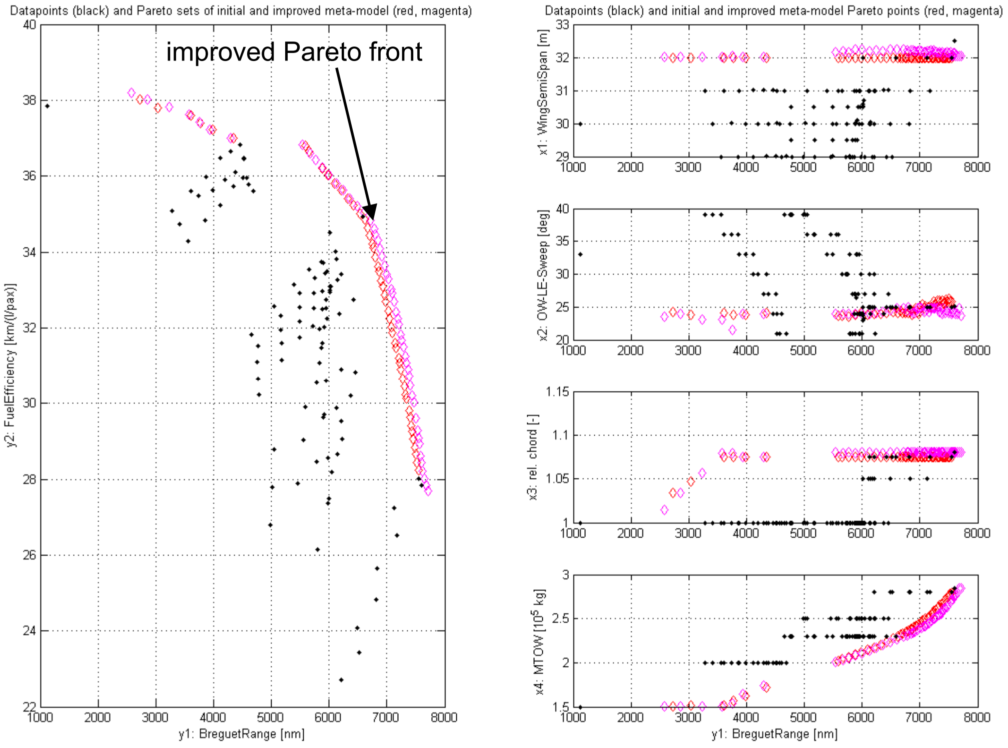


Fig. 16 Pareto fronts found with the initial meta-models (diamonds) and the improved meta-models (shifted diamonds).

The behavior of the aircraft in the parameter space around the Pareto optimal design points (which were predicted on the basis of the meta-models) was further explored and interpreted by aircraft design experts. One candidate optimal design point was selected (Fig. 15) and accurately evaluated by the MDA simulation system. The results from this evaluation are given in Table 3 and Fig. 16.

Table 3 MDA analysis result and meta-model prediction for the candidate optimal design point.

parameters				MDA analysis		meta model	
span	sweep	chord	MTOW	range	FuEff.	range	FuEff.
32.5	25.1	1.08	285000	7594.6	27.8	7761.9	28.4

When considering these results more closely, we can conclude from the MDA analysis results that this point is an additional Pareto optimal design point (Fig. 16). The meta-models predicted somewhat over-estimated values for range and fuel efficiency for this point (Table 3).

Furthermore, this new design point provides a valuable additional point for the data set on which the meta-models are created, and hence the meta-models can be further improved and used again in the multi-objective optimization. Therefore the meta-models for range and fuel efficiency were regenerated using the same kriging models as before (kcg for range and kle for fuel efficiency). In this optimization the 100 design points from the new data set are used as the initial generation and the bounds of the search domain are set to the minimum and maximum values of the design parameters of the 100 design points.

The resulting improved Pareto front (shifted diamonds in Fig. 16) found with these improved meta-models provides a slight improvement compared to the Pareto front (diamonds in Fig. 16) found with the previous meta-models, as is shown in Fig. 16. The Pareto front again helps to further guide the computationally expensive full MDA evaluations to the most interesting designs for the team of expert designers.

V Conclusions

The combination of advanced meta-models and multi-objective optimization algorithms for aircraft design presented in this chapter is flexible and applicable to a variety of design problems. A key benefit of this approach is that large numbers of interesting (Pareto optimal) design points can be found relatively quickly and easily at the cost of only few computationally expensive analyses, whilst a reasonable level of the accuracy is maintained. Representation of the results in the design parameter space as well as in the objective space provides valuable information for design decisions, where involvement of design specialists is required.

However, for high-dimensional design problems the visualization, assessment and selection of the most interesting design points require special attention. The same applies to the accuracy of the objective function values as predicted by the fits. Several ways to deal with the accuracy aspect were demonstrated:

- Use as much as possible information that is available, e.g. proper DOE, as many data points as possible, a priori knowledge of the underlying functions
- use different fitting methods and determine the best fit; carefully define appropriate validity domains for the fits (e.g. avoid extrapolation)
- try to account for fitting errors in the fit prediction by incorporating residual estimations.

The ε-NSGA-II algorithm has proven an efficient algorithm for calculating the Pareto optimal design points of the multi-objective design optimization, requiring only 20 seconds for this optimization run on a standard pc, which is very quick as compared to about 30 minutes computation time for a single MDA evaluation. The Pareto points based on the meta-model still need to be verified by the full MDA and validated by expert judgment. It was shown in the case study that, although the meta-models somewhat over-predicted the Pareto optimal points, the verified values are still on the Pareto front. From this verification the meta-model can be further improved and this contributes to further refinement of the optimization process.

The optimization approach followed here was also used in an engine design case [40], so is applicable to a larger set of design problems than only wing design.

References

[1] Pardessus T. Concurrent engineering development and practices for aircraft design at Airbus, Proc. 24th ICAS Conference, 2004, Yokohama, Japan, International Council of the Aeronautical Sciences (ICAS), Stockholm, Sweden, Optimage Ltd., Edinburgh, UK, 2004, paper nr. 2004-7.7.1.

[2] Torenbeek E. Synthesis of Subsonic Airplane Design. Delft University Press, Kluwer Academic Publishers, Delft, The Netherlands. ISBN 90-247-2724-3. pp. 215-262. 1982.

[3] Sobieszczanski-Sobieski J. and R. T. Haftka. Multidisciplinary aerospace design optimization: survey of recent developments, Structural and Multidisciplinary Optimization, Springer, Berlin/Heidelberg, Volume 14, Number 1, pp. 1 – 23, 1997.

[4] Laban M., P. Arendsen, W.F.J.A. Rouwhorst and W.J. Vankan. A Computational Design Engine for multi-disciplinary optimization with application to a Blended Wing Body configuration, Proc. AIAA/ISSMO Conference 2002, Atlanta, USA, September 2002.

[5] Ko A., L.T. Leifsson, W.H. Mason, J.A. Schetz, B. Grossman and R.T. Haftka. MDO of a Blended-Wing-Body transport aircraft with distributed propulsion, AIAA-2003-6732, Proc. AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Forum, Denver, Colorado, Nov. 17-19, 2003.

[6] Kesseler E., M. Laban and W.J. Vankan. Multidisciplinary wing optimization, VIVACE Forum-1, Warwick, UK, 20-21 September 2005; NLR-TP-2005-540.

[7] Simpson T.W., J. Peplinski, P.N. Koch and J.K. Allen. Metamodels for Computer-Based Engineering Design: Survey and Recommendations, Engineering with Computers, 17:2 (129-150), 2001.

[8] Zink P.S., D.A. DeLaurentis, M.A. Hale, V.V. Volovoi, D.P. Schrage, J.I. Craig, R.E. Fulton, F. Mistree, D.N. Mavris, W. Chen, P.J. Röhl, K.E. Lewis, P.N. Koch, C.E.S. Cesnik and T.W. Simpson. New Approaches to High Speed Civil Transport Multidisciplinary Design and Optimization, Proc. 2000 IEEE Aerospace Conference, Big Sky, Montana, March 18-25, 2000.

[9] DeLaurentis D.A., D.N. Mavris, and D.P. Schrage. System Synthesis in Preliminary Aircraft Design Using Statistical Methods, Proc. 20th ICAS Congress, Sorrento, Italy, September 8-13, pp. 866-878, 1996.

[10] Vankan W.J., W.F. Lammen, J. Kos and R. Maas. Complementary approximate modeling in Matlab and Modelica, Proc. Eurosim 2004 Conference, September 2004, Paris, France, 2004.

[11] Obayashi S., D. Sasaki, Y. Takeguchi, N. Hirose. Multiobjective Evolutionary Computation for Supersonic Wing-Shape Optimization. IEEE Transactions on Evolutionary Computation 4, 2 (July 2000), 182–187.

[12] Oyama, A. Multidisciplinary Optimization of Transonic Wing Design Based on Evolutionary Algorithms Coupled with CFD SOLVER. In Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering (2000).

[13] Oyama, A., and S. Obayashi. Multidisciplinary Wing Design Optimization Using Multiobjective Evolutionary Algorithm. In PPSN/SAB Workshop on Multiobjective Problem Solving from Nature (MPSN) (Faris, Prance, 2000).

[14] Oyama, A., S. Obayashi, K. Nakahashi and N. Hirose. Coding by Taguchi Method for Evolutionary Algorithms Applied to Aerodynamic Optimization. In Proceedings of the Fourth ECCOMAS Computational Fluid Dynamics Conference (Athens, Greece, September 1998), John Wiley & Sons, pp. 196–203, 1998, Vol. 2, K.D. Papailliou, D. Tsahalis, J. Périaux, D. Knörzer, Eds.

[15] Zitzler, E., M. Laumanns and S. Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. In Metaheuristics for Multiobjective Optimization (Berlin, Germany, 2004), Springer-Verlag, pp. 3–38.

[16] Deb, K. Multiobjective Optimization Using Evolutionary Algorithms. Wiley, Chichester, UK, pp. 77-300, 2001.

[17] Deb, K. Single and Multiobjective Optimization using Evolutionary Computation. KanGAL, Technical report 2004003, Feb. 2004.

[18] Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 2 (2002), 182–197.

[19] Deb, K., M. Mohan and S. Mishra. A Fast Multiobjective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions. KanGAL, Technical report 2003002, Feb. 2003.

[20] Fonseca, C. M., and P.J. Flemming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Proceedings of the Fifth International Conference on Genetic Algorithms (1993), S. Forrest, Ed., University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers, pp. 416–423.

[21] Knowles, J. D. and D.W. Corne. Approximating the Non-dominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 8, 2 (2000), 149–172.

[22] Zitzler, E., M. Laumanns and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems, Eurogen 2001 (Athens, Greece, 2001), CIMNE, pp. 95–100, 2002.

[23] Zitzler, E., and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and The Strength Pareto Approach. IEEE Transactions on Evolutionary Computation 3, 4 (1999), 257–271.

[24] The MathWorks. Optimization Toolbox http://www.mathworks.com/products/optimization.

[25] The MathWorks. Genetic Algorithm and Direct Search Toolbox http://www.mathworks.com/products/gads.

[26] Kleijnen J.P.C. and R.G. Sargent. A methodology for the fitting and validation of metamodels in simulation, European Journal of Operational Research, 120, no. 1, pp. 14-29, 2000.

[27] Bradley E. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation, Journal of the American Statistical Association, Vol. 78, No. 382, pp. 316-331, 1983.

[28] Lophaven, S.N., H.B. Nielsen and J. Søndergaard. DACE - A Matlab Kriging Toolbox, Version 2.0, Report IMM-REP-2002-12, Informatics and Mathematical Modelling, DTU, pp. 170-180, 2002.

[29] Bäck, T., U. Hammel and H.P. Schwefel. Evolutionary Computation: Comments on the History and Current State. IEEE Transactions on Evolutionary Computation 1, 1 (1997), 3–17.

[30] Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA, pp. 197-198, 1989.

[31] Zitzler, E. Evolutionary Algorithms, Multiobjective Optimization, and Applications. Tech. rep., Centro Internacional de Matematica, Portugal, 2003.

[32] Kollat, J.B. and P.M. Reed, The Value of Online Adaptive Search: A Performance Comparison of NSGAII, ε-NSGAII and εMOEA. In The Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Lecture Notes in Computer Science 3410 (386–398), 2005.

[33] Wolpert, D. H., and W.G. Macready. No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1, 1 (1997), 67–82.

[34] Deb, K., and R.B. Agrawal. Simulated Binary Crossover for Continuous Search Space. Complex Systems 9, 1995, 115–148 .

[35] Deb, K., and M. Goyal. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. Computer Science and Informatics 26, 4 (1996), 30–45.

[36] Kesseler E. and W.J. Vankan. Multidisciplinary design analysis and multi-objective optimization applied to aircraft wing, WSEAS transactions on systems and control, issue 2, volume 1, page 221 227, December 2006. NLR-TP-2006-748.

[37] Vankan, W.J., E. Kesseler and E.H. Baalbergen. Distributed collaborative and multi-disciplinary design of civil aircraft wings, PDT Europe 2006, October 16th-18th, Toulouse, France. NLR-TP-2007-168.

[38] Vankan W.J., R. Maas and M. Laban. Fitting fitness in aircraft design, ICAS 2006 Conference, September 2006, Hamburg, Germany, 2006. NLR-TP-2006-430.

[39] Bishop, C.M. Neural Networks for Pattern Recognition, Clarendon Press, Oxford (1997).

[40] Kesseler, E., M. Guenov, eds.: Advances in Collaborative Civil Aeronautical Multidisciplinary Design Optimization, AIAA, 2010.