Early Ideas and Innovations in Bayesian and Model-Assisted Multiobjective Optimization

Michael T.M. Emmerich

Faculty of Information Technology, University of Jyväskylä, Agora PL 35 40014 Jyväskylä, Finland
michael.t.m.emmerich@jyu.fi

Abstract. The early 2000s saw rapid growth in multiobjective design optimization, fueled by advances in evolutionary algorithms, Bayesian optimization, and surrogate models such as neural networks, Kriging, and Gaussian processes. These methods enabled efficient exploration of complex design spaces while reducing the cost of expensive function evaluations. Model-Assisted Multiobjective Optimization transformed the field by integrating surrogate-assisted techniques to enhance both efficiency and solution quality. This chapter reviews its historical development, emphasizing key contributions from European research efforts—including the European Community on Computational Methods in Applied Sciences and the INGENET network. We highlight foundational methodologies such as Pareto Efficient Global Optimization, Expected Hypervolume Improvement, and confidence-bound and interval-based pre-selection in Bayesian optimization. The chapter also examines the impact of these approaches on engineering applications, particularly in computational fluid dynamics and finite element method simulations, offering a comparative analysis of core concepts and their roles in shaping the landscape of modern multiobjective optimization.

Keywords: Multiobjective Optimization, Surrogate Modeling, Hypervolume Improvement, Engineering Design, Bayesian Optimization, Evolutionary Algorithms

1 Introduction

The early 2000s marked an explosion of ideas in multiobjective design optimization, fueled by the convergence of several key advancements: the emergence of multiobjective evolutionary algorithms (MOEAs) [49,8], the increasing adoption of Bayesian optimization for black-box problems [28,41], and the computational maturity of surrogate models—such as response surface models based on neural networks and Kriging (or Gaussian Process Regression) [45,32]—that could handle multidimensional and multimodal functions. Together, these innovations transformed the approach to complex optimization problems, shifting the paradigm from exhaustive brute-force methods to intelligent, model-assisted strategies.

In many engineering and scientific applications, optimization problems involve multiple conflicting objectives. Traditional solution methods relied on either exhaustive simulation-based approaches or heuristic search techniques, both of which are computationally intensive [26]. As simulation models grew more complex, particularly in areas like computational fluid dynamics (CFD) and finite element method (FEM) simulations [19], conventional optimization techniques struggled to cope with the high cost of function evaluations. This challenge led to the development of *Model-Assisted Multiobjective Optimization* (MAMO), a framework that integrates surrogate models to approximate expensive functions and efficiently explore the Pareto-optimal set [14].

By making use of data-driven approximations, MAMO reduces the number of costly function evaluations while maintaining high-quality approximations of Pareto-optimal solutions. Early research demonstrated that surrogate-assisted optimization could accelerate convergence and yield robust solutions across various engineering disciplines [36]. The integration of Kriging-based Gaussian processes [45], neural network surrogates, and Bayesian sampling strategies [37] allowed optimization algorithms to learn from previous evaluations, predict promising regions of the search space, and iteratively refine solutions.

The paper is organized as follows. In Section 2, we provide historical context by discussing early research initiatives that addressed computationally expensive design problems in CFD and FEM simulations [19]. Section 3 introduces key methodological foundations in optimization, covering single-objective and multiobjective optimization as well as surrogate modeling techniques, including Radial Basis Function Networks (RBFNs) [39] and Gaussian Process Regression (GPR) [45]. Section 4 surveys surrogate models in single-objective optimization.

The next three sections present foundational contributions to surrogate-assisted multiobjective optimization: Section 5 reviews the RBFN-based approach introduced by Giotis and Giannakoglou [22], Section 6 details the ParEGO method that employs a scalarization-based strategy with Gaussian Process models [30], and Section 7 discusses the Expected Hypervolume Improvement (EHVI) framework that targets Pareto front enhancement directly [44].

Sections 8, 9, and 10 discuss alternative infill criteria and early methods for using surrogate models: Section 8 examines non-dominated sorting of expected improvements (following the work of Jeong and Obayashi[25]), Section 9 describes the SMS-EGO framework that exploits lower confidence bounds, and Section 10 introduces interval filters that compare two-sided confidence intervals for candidate selection.

Section 11 provides a comparative analysis of these early contributions, highlighting their respective advantages and computational trade-offs. Finally, Section 12 concludes with a summary of findings and an outlook on future research directions. An appendix is provided with detailed examples, including a comprehensive Kriging prediction with confidence intervals.

2 Historical Context: European Initiatives in Multi-Objective Design Optimization

The early 2000s were transformative for multi-objective optimization in engineering design, driven in part by systematic efforts within the *European Community on Computational Methods in Applied Sciences* (ECCOMAS) and the *European Thematic Network on Intelligent Optimization and Inverse Problems in Engineering* (INGENET) [21,10]. These communities recognized that real-world engineering challenges—especially those involving CFD and FEM simulations—required optimization methods capable of handling computationally expensive, simulation-based black-box models.

A central objective of these efforts was to develop systematic benchmark suites that not only advanced optimization algorithms but also provided a common testing ground for evaluating methodologies in realistic engineering contexts. The INGENET network, in particular, played a key role in curating benchmark problems (see Appendix for two examples) that embodied the trade-offs inherent in multiobjective design, spanning areas such as aerodynamics, structural mechanics, and thermal systems [35]. These initiatives bridged applied mathematics and computational engineering, fostering interdisciplinary collaboration and accelerating innovation in real-world optimization.

During this period, other significant research forums emerged. The **Evolutionary Multi-Criterion Optimization** (EMO) conference series (inaugurated in 2001) consolidated both theoretical and applied advances in MOEAs, while the **Adaptive Design and Manufacture** (ACDM) conferences facilitated knowledge exchange on engineering design optimization, including pioneering work in multiobjective formulations. The **EUROGEN** conference series, organized under ECCOMAS, further served as a publication channel for early advances in surrogate-assisted methods and their practical applications.

A breakthrough concerning the enablement of multiobjective optimization in design engineering was the emergence of *Model-Assisted Multiobjective Optimization* (MAMO), which employed surrogate models to significantly reduce the computational burden of expensive multiobjective tasks. By constructing data-driven approximations of the objective functions, MAMO reduced the number of costly simulations required while still delivering high-quality approximations of the Pareto front. These early contributions laid the foundation for today's efficient, surrogate-based optimization techniques and have continued to influence applications ranging from digital twin technology to sustainability-driven engineering design.

3 Preliminaries on Optimization and Surrogate Models

Subsequently, we provide a brief introduction of the fundamental concepts that form the basis for the discussions in the following sections.

3.1 Single- and Multiobjective Optimization

Optimization involves finding the best solution from a set of feasible alternatives. In **single-objective optimization**, one seeks the optimal solution for a function $f(\mathbf{x})$:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}), \tag{1}$$

where $\mathcal{X}$ is the feasible search space and the optimal solution $\mathbf{x}^*$ satisfies

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{2}$$

A special case is direct or black-box optimization, where function evaluations are expensive (e.g., due to CFD or FEM simulations). In such settings, surrogate models (such as RBFNs) provide efficient approximations of these expensive evaluations and guide the search toward optimal solutions.

In **multiobjective optimization**, several conflicting objectives

$$f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ \ldots,\ f_m(\mathbf{x})$$

are optimized simultaneously:

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x}) = \big(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\big), \tag{3}$$

with no single solution simultaneously optimizing all objectives. Instead, one seeks the set of Pareto-optimal solutions, where a solution $\mathbf{x}^*$ is Pareto-optimal if there is no other $\mathbf{x} \in \mathcal{X}$ that improves one objective without worsening at least one other.

Radial Basis Function Networks (RBFNs) Radial Basis Function Networks (RBFNs) [3] are a class of artificial neural networks commonly used for function approximation. They typically have three layers: an input layer, a hidden layer where a nonlinear mapping is applied, and a linear output layer. For simplicity, we first consider the case of a single output ($l = 1$).

For the class of RBFN suggested by Giotis and Giannakoglou [22] the number of RBF centers equals the number of training points. During the training phase, the network learns to reproduce exactly the function values at known points. Unlike polynomial regression, it can reproduce exact data points from multimodal functions. When predicting the response at a new point $\mathbf{x} \in \mathbb{R}^n$, the network uses a weighted sum of radial basis functions centered at the training points. Emmerich [11] showed that this method produces equations much like those in Simple Kriging, which we will cover later.

RBFN Architecture Let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ denote the training points and let $y^{(i)} = y(\mathbf{x}^{(i)})$. For each training point, the RBF center is defined as:

$$\mathbf{b}^{(i)} := \mathbf{x}^{(i)}, \quad i = 1, \ldots, m. \tag{4}$$

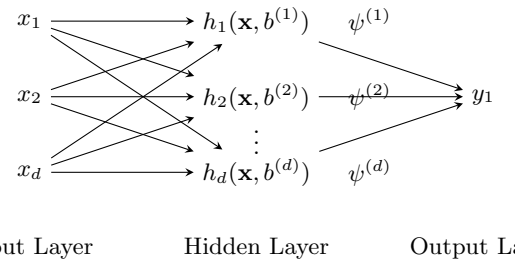

Fig. 1: Architecture of a Radial Basis Function Network (reproduced from [11]).

Using a norm $|\cdot|$ on $\mathbb{R}^n$ and a positive definite function $r : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, the activation function of the ith hidden unit is defined by:

$$h(\mathbf{x}, \mathbf{b}^{(i)}) := r\big(|\mathbf{x} - \mathbf{b}^{(i)}|\big), \quad i = 1, \ldots, m. \tag{5}$$

Giotis and Giannakoglou [22] suggest using

$$r(\mathbf{x}) = \exp\big(-|\mathbf{x} - \mathbf{x}'|^q\big) \quad \text{with } q = 2. \tag{6}$$

Alternatively, a weighted distance measure

$$|\mathbf{x} - \mathbf{x}'|_\theta = \sum_{i=1}^{n} \theta_i |x_i - x_i'|$$

can be used, with weights θ_i provided by the user or estimated from local gradients.

Input-Output Mapping The output of the RBFN is a linear combination of the hidden layer activations:

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{m} \psi^{(i)}\, h(\mathbf{x}, \mathbf{b}^{(i)}). \tag{7}$$

The weights $\psi^{(i)}$ are determined during training by requiring that the network exactly interpolates the training data:

$$\sum_{i=1}^{m} \psi^{(i)}\, h(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}) = y^{(j)}, \quad j = 1, \ldots, m. \tag{8}$$

In matrix form, this is written as:

$$\underbrace{\begin{bmatrix} h(\mathbf{x}^{(1)}, \mathbf{b}^{(1)}) & \cdots & h(\mathbf{x}^{(1)}, \mathbf{b}^{(m)}) \\ \vdots & \ddots & \vdots \\ h(\mathbf{x}^{(m)}, \mathbf{b}^{(1)}) & \cdots & h(\mathbf{x}^{(m)}, \mathbf{b}^{(m)}) \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \psi^{(1)} \\ \vdots \\ \psi^{(m)} \end{bmatrix}}_{\boldsymbol{\psi}} = \underbrace{\begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}}_{\mathbf{y}}. \tag{9}$$

Assuming no duplicate training points and positive definite r, the solution is

$$\boldsymbol{\psi} = \mathbf{H}^{-1}\mathbf{y}. \tag{10}$$

3.2 Kriging and Gaussian Process Regression

Kriging is a statistical interpolation method used for spatial prediction and function approximation. It was developed by Matheron and was inspired by the initial ideas of Krige, the mining engineer [32,36].

The Kriging method constructs an estimate of an unknown function at a new point by forming a weighted combination of known values while taking spatial correlation into account. In addition to providing a linear predictor, Kriging also supplies an uncertainty measure, often expressed as mean squared error.

For a set of sample points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}$ with corresponding values $y^{(1)}, y^{(2)}, \ldots, y^{(m)}$, the prediction at a new point $\mathbf{x}'$ is given by:

$$\hat{y}(\mathbf{x}') = \beta + \sum_{i=1}^{m} \lambda^{(i)}\, c\big(\mathbf{x}', \mathbf{x}^{(i)}\big), \tag{11}$$

where:

- β is the known mean in Simple Kriging,
- $\lambda^{(i)}$ are weights determined from the covariance function,
- $c(\mathbf{x}', \mathbf{x}^{(i)})$ is the covariance (or correlation) between $\mathbf{x}'$ and $\mathbf{x}^{(i)}$.

The weights $\lambda^{(i)}$ are computed by solving

$$\mathbf{C}\,\boldsymbol{\lambda} = \mathbf{c}(\mathbf{x}'), \tag{12}$$

with $\mathbf{C}$ being the covariance matrix whose entries are $C_{ij} = s(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ and $\mathbf{c}(\mathbf{x}')$ the covariance vector between $\mathbf{x}'$ and each training point.

It is important to note that

- the covariance function is estimated from the data.
- The statistical model quantifies prediction uncertainty using variance $\hat{s}(\mathbf{x}')$, which depends on proximity to data points and can be interpreted as the prediction's mean squared error (MSE) via the BLUP.

In the appendix we provide an example with a more details.

Simple Kriging In Simple Kriging, the mean β is assumed known and constant over the domain, simplifying the prediction to:

$$\hat{y}(\mathbf{x}') = \beta + \sum_{i} \lambda_i\, (y^{(i)} - \beta). \tag{13}$$

While this assumption works well when the mean is indeed constant, it is less flexible if the mean varies spatially.

Gaussian Process Regression (GPR) Gaussian Process Regression (GPR) [45] is closely related to Kriging. In GPR, the prediction at a new point is given by the mean of the predictive posterior distribution, and the associated variance quantifies uncertainty. Although both methods yield a predictor and an uncertainty measure, GPR is embedded in a full probabilistic framework that includes priors and Bayesian inference.

3.3 Relation between Kriging and RBFNs

RBFNs and Simple Kriging share a common idea: both construct predictions as a linear combination of basis functions centered at the training points. Table 1 summarizes the correspondences between the two approaches.

Simple Kriging	**RBFN**
Prediction: $\hat{y}(\mathbf{x}') = \beta + \sum \lambda_i \, c\big(\mathbf{x}', \mathbf{x}^{(i)}\big)$	Output: $\hat{y}(\mathbf{x}') = m + \sum w_i \, h\big(\mathbf{x}', \mathbf{x}^{(i)}\big)$
Spatial indices: $\mathbf{x}^{(i)}$	RBF centers: $\mathbf{x}^{(i)}$
Best linear unbiased predictor (BLUP)	Predicted output
Covariance function $c(\mathbf{x}', \mathbf{x}^{(i)})$	Activation function $h\big(\mathbf{x}', \mathbf{x}^{(i)}\big)$
Weights λ_i	Weights w_i
Variance $\hat{s}$	No uncertainty quantification
Correlation parameters adapted to data	Parameters of activation function fixed

Table 1: Comparison Simple Kriging vs. Radial Basis Function Networks.

The activation function in RBFNs corresponds to the positive definite correlation function in Kriging. Thus, when RBFNs are used for interpolation, the resulting equations are equivalent to those of Simple Kriging.

4 Surrogate Models in Single-Objective Optimization

The use of surrogate models in single-objective optimization has a long history, with many ideas later generalized to multiobjective settings (e.g., Bayesian optimization and expected improvement [37,38]). In particular, the work of Yaochu Jin on data-driven approximations [26] greatly influenced single-objective evolutionary optimization. We briefly introduce two archetypal algorithms.

4.1 Bayesian Optimization for a Single-Objective Problem

Bayesian optimization, originally introduced by Mockus et al. [37,38], is a model-based approach designed for optimizing expensive black-box functions. It is widely used in hyperparameter tuning, engineering design, and scientific optimization problems where function evaluations are costly. The key idea is to maintain a probabilistic surrogate model (often a Gaussian Process) that provides both predictions and uncertainty estimates. An acquisition function is then used to balance exploration and exploitation by deciding where to sample the next point.

$$EI(\mathbf{x}) = \begin{cases} \big(\hat{y}^* - \hat{y}(\mathbf{x})\big)\, \Phi\!\left(\dfrac{\hat{y}^* - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x})\, \phi\!\left(\dfrac{\hat{y}^* - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right), & \hat{s}(\mathbf{x}) > 0, \\ 0, & \hat{s}(\mathbf{x}) = 0. \end{cases} \tag{14}$$

This value can be interpreted as the expected progress in the direction of the global optimum, given the observed function evaluation are a sample of a random field indexed by the input variable vector.

Algorithm 1 Bayesian Optimization for a Single-Objective Problem

Initialize: Choose a surrogate model S (e.g., a Gaussian Process).
Collect an initial dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ by evaluating $f(\mathbf{x})$ at a set of initial points.
for $t = 1$ to max_iterations **do**
    Train the surrogate model S using $\mathcal{D}$.
    Select the next evaluation point:

$$\mathbf{x}_{\text{new}} = \arg\max_{\mathbf{x}} \text{AcquisitionFunction}(S, \mathcal{D})$$

    Evaluate the expensive objective: $y_{\text{new}} = f(\mathbf{x}_{\text{new}})$
    Update the dataset: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_{\text{new}}, y_{\text{new}})\}$
    if stopping criterion is met **then**
        Break
    end if
end for
return Best solution found.

The algorithm, summarized in the pseudocode of Algorithm 1, works as follows: In the beginning one selects a surrogate model (for example, a Gaussian Process) and evaluates the objective function at some initial points. Then one fit the surrogate model S on the current dataset $\mathcal{D}$ to train it well. Next, one uses an acquisition function (for example, Expected Improvement or Upper Confidence Bound) to choose the next point to evaluate, which is very important to guide where to search next. After that, one evaluates the objective function at the selected point and update the dataset $\mathcal{D}$ accordingly. Finally, the process continues until a stopping criterion, like a budget or convergence, is met, and when no further improvement is observed, one stops.

This approach was later popularized under the name "Efficient Global Optimization (EGO)" after a publication by Jones et al. [28] which revived interest in the approach - a term that, while widely used, is somewhat of a misnomer since it does not guarantee efficiency if the evaluation budget is known a priori [48].

Note that the term "Bayesian Optimization" is also used to refer to estimation of distribution algorithms, which model populations in evolutionary algorithms through stochastic distributions [34]. However, these methods are not specifically designed for problems with expensive function evaluations.

4.2 Surrogate-Assisted Evolutionary Algorithm with Pre-Selection

Evolutionary algorithms generate new offspring by applying mutation and recombination operators to a population of candidate solutions. In surrogate-

assisted variants [19], machine learning models are used to reduce the number of expensive function evaluations. In pre-selection strategies, offspring candidates are first evaluated using the surrogate model, and only the most promising ones are subsequently evaluated with the true objective function.

Early approaches ranged from simple neural network predictions [27] to sophisticated methods incorporating uncertainty quantification via Kriging. Notable examples include metamodel-assisted evolution strategies using lower confidence bounds [15], the parameterless probability of improvement [43], and variants using Expected Improvement together with constraint-handling strategies [11,14].

Algorithm 2 Model-Assisted Evolutionary Algorithm with Pre-Selection

Initialize: Generate an initial population P of random individuals.
Train a surrogate model S using initial evaluations.
for $t = 1$ to max_generations **do**
    Generate an offspring population Q via variation operators (mutation/crossover).
    Evaluate each $\mathbf{x} \in Q$ using the surrogate model:

$$\{\hat{y}(\mathbf{x}), \hat{s}(\mathbf{x}) \mid \mathbf{x} \in Q\}$$

    Select a subset $Q' \subseteq Q$ of promising candidates based on $\hat{y}$.
    Evaluate the true objective for individuals in Q'.
    Update the surrogate model S with the new evaluations.
    Form the next generation population P via selection
     (e.g., comma-selection or plus-selection).
    if stopping criterion is met **then**
        Break
    end if
end for
return Best solution found.

The main loop of a model-assisted evolutionary algorithm is shown in Alg. 2. It begins with the initialization of a population by randomly sampling candidate solutions. A surrogate model is then trained using a limited number of true evaluations to approximate the objective function. Offspring are generated through mutation and crossover, creating new candidate solutions. To reduce computational costs, a pre-selection step utilizes the surrogate model to identify promising candidates. These selected candidates are then evaluated using the actual objective function, and the surrogate model is updated accordingly. Finally, a selection mechanism determines the next generation, ensuring iterative refinement of the population over successive iterations.

Evolutionary algorithms were quite popular in the early attempts to utilize surrogate models, but seem to recently receive less attention. However, there is notable work on the introduction of multiobjective optimization based on decomposition [5,23].

A conceptually similar alternative to model-assisted evolutionary optimization is model-assisted pattern search [42]. Unlike evolutionary algorithms that rely on stochastic sampling, model-assisted pattern search employs a deterministic sampling strategy. This approach enables the establishment of local convergence guarantees.

5 Giotis and Giannakoglou's RBFN-Based Method

Giotis and Giannakoglou [20,22] were early pioneers in the integration of surrogate models with evolutionary algorithms for multi-objective optimization. Focusing on aerodynamic design optimization, where objective evaluations involved costly CFD simulations, they employed response surfaces generated via polynomial regression and RBF-based neural networks. By integrating these surrogates with evolutionary algorithms, their work demonstrated that surrogate-supported optimization could effectively address complex engineering challenges. A special feature of this approach is the computation of importance factors as an interpretation of RBF weights. In hindsight, this technique might be viewed as an early eXplainable artificial intelligence (XAI) approach [2].

Key features

Giotis and Giannakoglou pioneered the integration of surrogate models with evolutionary algorithms for multi-objective aerodynamic design optimization by approximating costly CFD simulations with response surfaces generated through RBF-based neural networks. Their approach not only reduced computational costs in complex engineering problems but also featured the computation of importance factors derived from RBF weights, offering an early demonstration of model interpretability that anticipated modern explainable artificial intelligence.

6 ParEGO: Pareto-based Efficient Global Optimization

ParEGO, introduced by Knowles in 2006 [30,31], transformed multiobjective optimization into a series of scalarized single-objective problems. By systematically varying the weights in a Chebyshev aggregation function, ParEGO explores trade-offs among objectives. A Gaussian Process (or Kriging) model is used to quantify uncertainty, and the Expected Improvement acquisition function is applied to the scalarized objective.

The weighted Chebyshev scalarization function with the ideal point as reference is defined as:

$$g(\mathbf{x}, \mathbf{w}, \mathbf{z}^*) = \max_{i=1,\ldots,m} w_i \left| f_i(\mathbf{x}) - z_i^* \right|,$$

where:

- $\mathbf{x} \in \mathbb{X}$ is the decision vector,

- $\mathbf{w} = (w_1, \ldots, w_m)$ is a weight vector with $w_i > 0$ and $\sum_{i=1}^{m} w_i = 1$,
- $\mathbf{f}(\mathbf{x}) = \big(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})\big)$ is the objective vector,
- $\mathbf{z}^* = \big(z_1^*, \ldots, z_m^*\big)$ is the ideal point with $z_i^* = \min_{\mathbf{x} \in \mathbb{X}} f_i(\mathbf{x})$.

This scalarization converts the multiobjective problem into a single-objective one:

$$\min_{\mathbf{x} \in \mathbb{X}} g(\mathbf{x}, \mathbf{w}, \mathbf{z}^*),$$

allowing ParEGO to approximate the Pareto front by varying the weight vector $\mathbf{w}$. Notably, the Chebyshev scalarization can handle concave Pareto fronts more effectively than additive or linear scalarizations.

Key Features

ParEGO uses Gaussian Process models for surrogate modeling, iteratively updating with new samples to efficiently explore and converge to the Pareto-optimal set. It transforms the problem into single-objective tasks via weighted Chebyshev scalarization and calculates expected improvement. Its computational efficiency and broad applicability have made it widely adopted.

7 Expected Hypervolume Improvement (EHVI)

The Expected Hypervolume Improvement (EHVI) criterion extends the concept of expected improvement from single-objective to multiobjective settings by assessing the improvement of an approximation set toward the Pareto front as a whole. EHVI interprets improvement as an multidimensional integral the non-dominated space, where the weighting is derived from a probability density function. (see Fig. 2) Emmerich et al. (2006) proposed an EHVI framework that directly targets improvements in the hypervolume indicator, which measures the dominated volume in the objective space. Although EHVI was initially presented alongside other acquisition functions, it later gained prominence, especially after its successful application in Bayesian optimization schemes [18,33,44].

Early implementations of EHVI were computationally demanding due to expensive integral evaluations and reliance on Monte Carlo approximations. These challenges led to the development of more efficient exact evaluation schemes [6,24], culminating in algorithms with asymptotically optimal $O(n \log n)$ runtime for two and three objectives [18,46]. These improvements have made EHVI a practical and powerful tool for high-dimensional MAMO problems.

Note that an earlier approach to computing the expected improvement for Pareto front approximation with two objective functions was proposed by Keane. This method calculates the center of gravity over the dominated region using a measure that rewards filling gaps in the Pareto front approximation [29]. In [44] it has been compared to the EHVI.

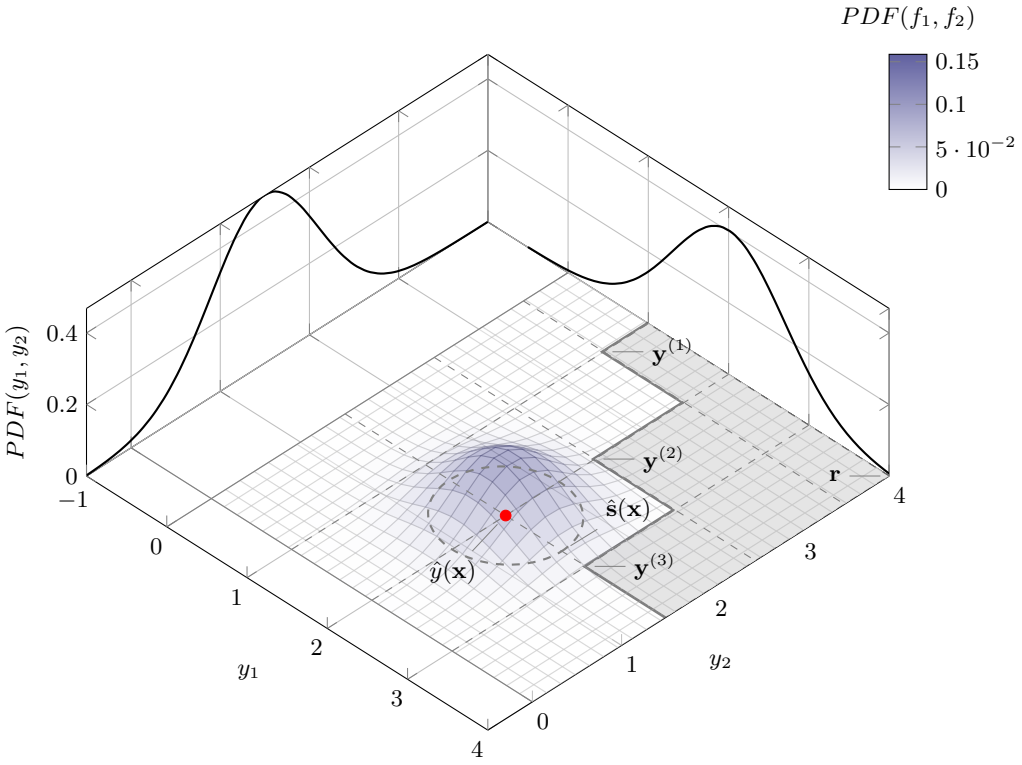

Fig. 2: Pareto front approximation and 2-D predictive distribution. The incumbent Pareto front is provided by $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{y}^{(3)}\}$. The grey area can be interpreted as the hypervolume indicator. The red point marks $\hat{y}(\mathbf{x})$ (mean), and the dashed ellipse represents $\hat{\mathbf{s}}(\mathbf{x})$ (variance) below the predictive surface.

Key Features

EHVI directly rewards improvements in the hypervolume indicator, thus explicitly targeting the Pareto front. Its reliance on a Lipschitz continuous, Pareto-compliant indicator helps ensure convergence, and recent algorithmic advances have mitigated its initial computational drawbacks.

8 Non-Dominated Sorting of Expected Improvement (Jeong & Obayashi)

Jeong and Obayashi [25] extended the Expected Improvement (EI) concept to multiobjective optimization by applying non-dominated sorting to the EI values computed for individual objectives. Instead of aggregating objectives through

scalarization or hypervolume measures, their method computes the EI for each objective separately and ranks candidate solutions based on Pareto dominance.

This approach naturally produced a well-distributed set of solutions and was computationally efficient owing to the closed-form expression of the single-objective EI. However, challenges such as correlations between EI values and efficient classification hindered their early adoption. Recent advances in multiobjective Bayesian optimization have revisited these principles, incorporating faster sorting techniques and uncertainty-aware strategies.

Key Features

By computing the EI separately for each objective and using non-dominated sorting, this approach avoids the pitfalls of scalarization and hypervolume computation. Its main advantage is computational efficiency, although the early versions did not explicitly address crowding distances or correlations among objectives.

9 SMS-EGO: Exploiting the Lower Confidence Bound

The lower confidence bound of a Kriging or GP prediction

$$\mathrm{lb}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x}) \tag{15}$$

has been used in single objective optimization as a infill criterion [7], but can be generalized in a straightforward way to multiobjective optimization by using a vector of independent predictions of objective function values. The factor ω can be used to scale between exploration and exploitation.

The idea of using hypervolume improvement based on a lower confidence bound within the model-assisted evolutionary algorithm was introduced in the seminal work on the SMS-EMOA multiobjective optimization algorithm. The hypervolume indicator offered a straightforward way to define improvement of a Pareto front approximation [13]. Applied initially to two problems from the INGENET benchmark suite (e.g., the NACA RAE 2822 drag minimization test case), the method showed promise for surrogate-assisted multi-objective optimization.

Within the Bayesian Optimization scheme, the approach of using lower bounds was successfully introduced by Ponweiser et al. Ponweiser et al. [40]. A limitation noted by Ponweiser et al. [40] was that the method struggled to detect improvements in dominated regions. To address this, the concept of negative hypervolume improvement was introduced and later refined in [44]. The resulting SMS-EGO framework combines Bayesian optimization with hypervolume-based selection and draws inspiration from both SMS-EMOA and EGO. Although the initial computational advantage of SMS-EGO over EHVI has diminished as exact EHVI computation methods improved, the idea of incorporating negative hypervolume improvement continues to influence recent developments, such as gradient-based hypervolume optimization [9].

Key Features

SMS-EGO leverages a lower confidence bound to extend single-objective strategies to a multiobjective context by applying independent predictions for each objective. The parameter ω balances exploration and exploitation, while the use of hypervolume improvement—including its negative variant—provides a direct measure for advancing Pareto front approximations, particularly in dominated regions. By combining Bayesian optimization with hypervolume-based selection, SMS-EGO integrates the strengths of SMS-EMOA and traditional EGO into a robust framework for surrogate-assisted multiobjective optimization.

10 Interval Filters

Thus far, the methods described have relied on scalar acquisition criteria. In contrast, interval filters compare multi-dimensional two-sided confidence intervals assigned to each candidate solution. These filters use possibilistic logic to sort, accept, or reject points for evaluation based on the entire confidence interval rather than a single scalar value.

Emmerich et al. [16] proposed defining lower and upper confidence bounds for each solution as:

$$\mathrm{lb}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) - \omega \cdot \hat{s}(\mathbf{x}), \quad \mathrm{ub}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x}) + \omega \cdot \hat{s}(\mathbf{x}). \tag{16}$$

Here, $\hat{y}(\mathbf{x})$ is the predicted value, $\hat{s}(\mathbf{x})$ is the associated uncertainty, and ω is a confidence parameter controlling the interval width. Under the Gaussian assumption, the probability that the true value falls within this interval is given by

$$p_\alpha = \Phi(\omega) - \Phi(-\omega), \tag{17}$$

with the convention that if $\hat{s}(\mathbf{x}) = 0$, then $\mathrm{lb}_\omega(\mathbf{x}) = \mathrm{ub}_\omega(\mathbf{x}) = \hat{y}(\mathbf{x})$.

When selecting candidate solutions for evolutionary algorithms, two types of errors may occur:

(A) Selecting solutions that are not among the top μ candidates.
(B) Rejecting solutions that belong to the top μ candidates.

A filter with high *precision* minimizes error (A), while one with high *recall* minimizes error (B). In multiobjective settings (minimization), the confidence intervals are depicted as orthogonal ranges in the objective and constraint spaces (see Figure 3). A solution is *possibly non-dominated* if its lower bound is not dominated with respect to the upper bounds of all other solutions (we can create a witness realization then where the solution is non-dominated) and *certainly dominated* if its lower bound is dominated by a upper bound of at least one other solution. A detailed discussion of interval filters can be found in [11].

These methods can also be extended to constrained optimization, as shown in [17] for a three-objective airfoil optimization test case. By training a surrogate model for the constraint boundary, lower and upper bounds for the constraint can be obtained, which then inform the acceptance or rejection of candidate solutions.

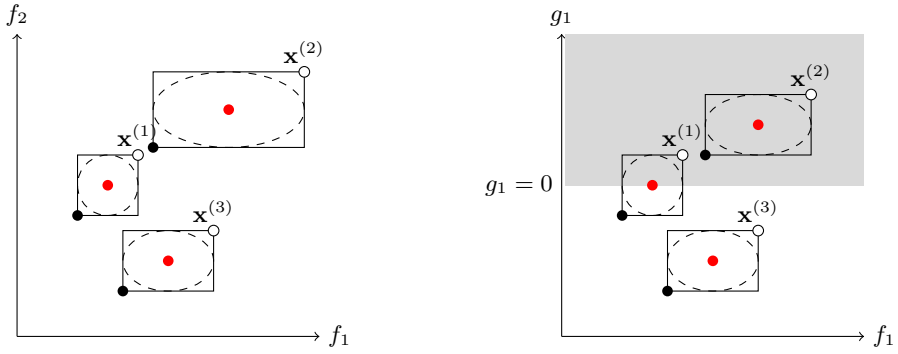

Fig. 3: Interval Filters for multiple objectives (a) and for implicit constraints (b), see also: [17].

Key Features

Interval filters assess candidate solutions using multi-dimensional two-sided confidence intervals rather than single scalar values. Each solution is assigned lower and upper confidence bounds—derived from its predicted value and uncertainty with a tunable width parameter—providing a probabilistic guarantee (under a Gaussian assumption) that the true value lies within the interval. This method effectively balances the risk of including inferior solutions and excluding top candidates, and in multiobjective settings, it differentiates possibly non-dominated solutions from those that are certainly dominated. Moreover, the approach extends naturally to constrained optimization by integrating surrogate models for constraint boundaries.

11 Comparison of Early Contributions and Ideas

Tables 2 and 3 compare the key aspects of these foundational methods. It becomes apparent that Kriging models (or Gaussian process models) became the prevalent choice, when it comes to surrogate models. It is interesting to note that Lipschitz models might be an interesting alternative approach, as outlined in a new article by Žilinskas [47].

Moreover, many ideas for acquisition functions can be used in both Evolutionary and Bayesian Optimization, and have counterparts from single-objective optimization. The only approach that does not make use of acquisition functions is the interval filters.

12 Conclusions and Outlook

The foundational contributions to Model-Assisted Multiobjective Optimization (MAMO) have paved the way for contemporary surrogate-assisted techniques.

Table 2: Comparison of Selected Early Contributions to MAMO

Aspect	**Giotis & Giannakoglou**	**ParEGO**	**EHVI**
Surrogate Model	Radial Basis Functions	Kriging/GP (scalarized)	Kriging/GP (per objective)
Optimization Strategy	Evolutionary Algorithms	Bayesian Optimization	Indicator-based EA, Bayesian Optimization
Uncertainty Quantification	–	Variances of scalarized improvement	Variance of hypervolume improvement
Applications	CFD optimization	Broad MAMO problems	Diverse multiobjective problems

Table 3: Comparison of Additional Early Contributions to MAMO

Aspect	**Jeong & Obayashi**	**SMS-EGO**	**Interval Filters**
Surrogate Model	Kriging/GP	Kriging/GP	Kriging/GP (per objective)
Optimization Strategy	NSGA-II	Bayesian Optimization	Evolutionary Algorithms
Uncertainty Quantification	Single-objective EI	Lower confidence bound	Upper and lower confidence bounds
Applications	CFD/airfoil optimization	Broad MAMO problems	CFD/airfoil optimization

This paper has reviewed key early methods—including those by Giotis and Giannakoglou, ParEGO, EHVI, SMS-EGO, and interval-based filters—that introduced diverse strategies for balancing exploration and exploitation and reducing computational costs in multiobjective settings.

The pioneering integration of RBFNs with evolutionary algorithms by Giotis and Giannakoglou demonstrated the feasibility of surrogate-assisted optimization for aerodynamic design. ParEGO's scalarization approach provided an effective framework using Gaussian Process models, while EHVI focused on direct hypervolume improvement of the Pareto front. Subsequent methods, such as SMS-EGO and non-dominated sorting of EI, further refined surrogate-based selection. Interval-based methods introduced robust uncertainty handling, which influenced subsequent developments in Bayesian optimization.

The field of MAMO has been advancing a lot over recent years and parallel computing (batch evaluation) possibilities as well as advances in machine learning were driving forces. Moreover, infill criteria have been revised and/or thoroughly tested and integrated into programming libraries. For more recent overview papers see [1,4].

Looking ahead, MAMO continues to evolve with advances in deep learning surrogates, transfer learning, and adaptive sampling strategies. These developments promise to further enhance the applicability of surrogate-assisted optimization in engineering, environmental planning, and even combinatorial fields such as model-driven drug discovery. The integration of scalable, model-aware optimization frameworks will be essential in addressing increasingly complex multi-objective problems.

A Kriging Prediction with Confidence Intervals: A Detailed Example

This appendix illustrates the computational process for using Kriging to predict values at unobserved locations. Although the example is for one-dimensional inputs, the method extends naturally to higher dimensions by adjusting the correlation function.

A.1 Mathematical Background

The Kriging prediction at a new point x is given by:

$$\hat{y}(x) = \beta + \mathbf{k}(x)^T \mathbf{K}^{-1}(\mathbf{y} - \beta\,\mathbf{1}),$$

where β is the global mean of the observed values, $\mathbf{y}$ is the vector of observations, and $\mathbf{1}$ is a vector of ones.

The variance of the prediction at x is:

$$\sigma^2(x) = \sigma_0^2 \left(1 - \mathbf{k}(x)^T \mathbf{K}^{-1} \mathbf{k}(x) + \frac{\left(1 - \mathbf{1}^T \mathbf{K}^{-1} \mathbf{k}(x)\right)^2}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}}\right),$$

with σ_0^2 representing the process variance.

A.2 Example with Three Data Points

Consider data points $x_1 = 1$, $x_2 = 2$, $x_3 = 3$ with observed values $y_1 = 1.0$, $y_2 = 2.0$, $y_3 = 1.5$. Assume a correlation function with parameter $\theta = 1$ and exponent $q = 2$.

Correlation Matrix The correlation matrix $\mathbf{K}$ is

$$\mathbf{K} = \begin{bmatrix} 1 & e^{-1} & e^{-4} \\ e^{-1} & 1 & e^{-1} \\ e^{-4} & e^{-1} & 1 \end{bmatrix}.$$

Its inverse $\mathbf{K}^{-1}$ is computed numerically.

Correlation Vector for $x = 2.5$ The correlation vector $\mathbf{k}(2.5)$ is

$$\mathbf{k}(2.5) = \begin{bmatrix} e^{-2.25} \\ e^{-0.25} \\ e^{-0.25} \end{bmatrix}.$$

Prediction and Variance The predicted value at $x = 2.5$ is

$$\hat{y}(2.5) = \beta + \mathbf{k}(2.5)^T \mathbf{K}^{-1}(\mathbf{y} - \beta\,\mathbf{1}),$$

with β typically set as the mean of the observations (here, 1.5). The prediction variance is computed as

$$\sigma^2(2.5) = \sigma_0^2 \left(1 - \mathbf{k}(2.5)^T \mathbf{K}^{-1} \mathbf{k}(2.5) + \frac{\left(1 - \mathbf{1}^T \mathbf{K}^{-1} \mathbf{k}(2.5)\right)^2}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}}\right).$$

A.3 Python Implementation

A Python implementation (see [12]) uses these formulas to calculate the predictions and uncertainties. By changing parameters (e.g., setting $q = 1$ instead of $q = 2$), one can study the impact on predictions. The figure below shows an example output with confidence intervals.

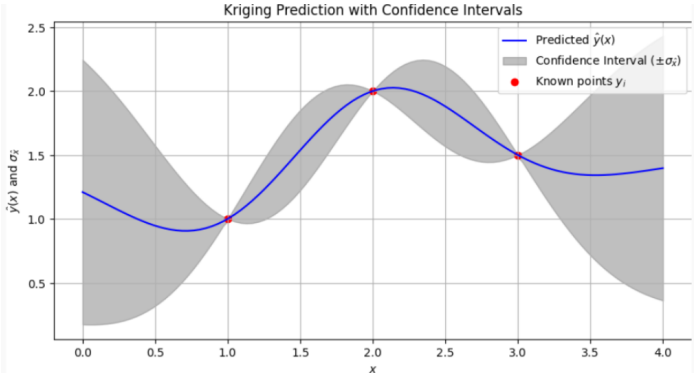

Fig. 4: Kriging prediction with confidence intervals.

B Key Terminology

– **Bayesian Optimization**: An optimization method using a statistical model-based acquisition function to guide evaluations of the objective function. The term is also used as a synonym for estimation of distribution algorithms in evolutionary compuation.

- **CFD Optimization**: The use of optimization techniques to improve aerodynamic performance.
- **Confidence Intervals in Surrogates**: Quantification of uncertainty in surrogate model predictions.
- **EHVI**: Expected Hypervolume Improvement, an acquisition function in Bayesian multiobjective optimization.
- **Expected Improvement (EI)**: An acquisition function that quantifies potential improvement in Bayesian optimization.
- **FEM**: Finite Element Method, a numerical technique in engineering design.
- **Gaussian Process Regression (GPR)**: A nonparametric regression method with strong conceptual similarities with Kriging, but focusing more on Bayesian interpretation of the prediction and variance and not less on linear estimator theory.
- **Hypervolume Indicator**: A metric that measures the volume dominated by a Pareto front.
- **Hypervolume-Based Infill Criterion**: A strategy that uses hypervolume improvement for candidate selection.
- **INGENET**: A European network focused on intelligent optimization methods.
- **Interval Filters**: Methods that compare confidence intervals to decide on candidate evaluation.
- **Kriging**: A surrogate modeling technique that provides predictions with uncertainty estimates.
- **Lower Confidence Bound (LCB)**: A statistic used to guide exploration in Bayesian optimization.
- **Metamodeling**: The use of surrogate models to approximate expensive simulations.
- **MAMO**: Model-Assisted Multiobjective Optimization.
- **Model-assisted Pattern Search**:
- **Multiobjective Optimization (MOO)**: The simultaneous optimization of multiple conflicting objectives.
- **Non-dominated Sorting**: A method for ranking solutions based on Pareto dominance.
- **NSGA-II**: A popular evolutionary algorithm for multiobjective optimization.
- **Pareto Front**: The set of non-dominated solutions in multiobjective optimization.
- **Radial Basis Function Networks (RBFN)**: Neural networks used for regression and multivariate function interpolation.
- **Scalarization**: The conversion of a multiobjective problem into a single-objective one.
- **Simple Kriging**: A form of Kriging with a known constant mean.
- **SMS-EGO**: A framework combining Bayesian optimization with hypervolume-based selection.
- **Surrogate-Assisted Evolutionary Algorithm (SA-EA)**: An evolutionary algorithm augmented with surrogate models.

C Summary of Airfoil Redesign Test Cases

Two airfoil redesign test cases from the INGENET database [10] served as benchmarks for early MAMO methods.

NACA Test Case

The goal is to compare an optimized airfoil with **NACA 0012** and **NACA 4412** under two flow conditions: take-off (high lift) and cruise (low drag). The objective functions are based on differences in the pressure distribution, and the test case features 18 degrees of freedom.

Condition	**Take-off**	**Cruise**
Mach number	0.20	0.77
Reynolds number	5×10^6	10^7
Angle of attack (°)	10.8	1.0
Transition	3%	3%

Table 4: Flow conditions for the NACA test case.

RAE Drag Minimization Test Case

For the RAE 2822 airfoil, the goal is to minimize drag coefficients over three flow conditions while satisfying lift and pitch moment constraints. Geometric constraints ensure appropriate airfoil thickness, leading-edge radius, and trailing-edge angle.

Condition	**Cruise**	**Off-design 1**	**Off-design 2**
Mach number	0.734	0.754	0.680
Reynolds number	6.5×10^6	6.2×10^6	5.7×10^6
Angle of attack (°)	2.8	2.8	1.8
Transition	3%	3%	11%

Table 5: Flow conditions for the RAE test case.

In the RAE test case, the objectives are to optimize the drag coefficient while maintaining sufficient lift and keeping the pitching moment within a 2% range.

References

1. R. Allmendinger, M. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni. Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case. *Journal of Multi-Criteria Decision Analysis*, 24(1-2):5–24, 2017.
2. LR Biggers, C Bocovich, R Capshaw, BP Eddy, LH Etzkorn, NA Kraft, et al. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *Empirical Softw. Eng*, 19(3), 2014.
3. D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2(3):321–355, 1988.
4. T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya. A surrogate-assisted evolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 21(5):739–752, 2016.
5. T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):129–142, 2016.
6. I. Couckuyt, D. Deschrijver, and T. Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. *Journal of Global Optimization*, 60:575–594, 2014.
7. D.D. Cox and S. John. A statistical method for global optimization. In *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE, 1992.
8. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
9. T.M. Deist, S.C. Maree, T. Alderliesten, and P.A.N. Bosman. Multi-objective optimization by uncrowded hypervolume gradient ascent. In *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part II 16*, pages 186–200. Springer, 2020.
10. J.-A. Désidéri and N. Marco-Blaszka. INGENET test-case database: structure and examples of optimization problems. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2000)*, pages 1–20, Barcelona, Spain, September 2000. 11–14 September 2000.
11. M. Emmerich. *Single- and Multi-Objective Evolutionary Design Optimization Assisted by Gaussian Random Field Metamodels.* PhD thesis, University of Dortmund, 2005.
12. M. Emmerich. Python implementation of dynamic programming for minimum riesz s-energy subset selection in ordered point sets. Zenodo, 10.5281/zenodo.14792491, 2025.
13. M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76. Springer, 2005.
14. M. Emmerich, K.C. Giannakoglou, and B. Naujoks. Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.
15. M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou. Metamodel—assisted evolution strategies. In *International Conference on Parallel Problem Solving from Nature*, pages 361–370. Springer, 2002.

16. M. Emmerich and B. Naujoks. Metamodel assisted multiobjective optimisation strategies and their application in airfoil design. In *Adaptive computing in design and manufacture VI*, pages 249–260. Springer, 2004.
17. M. Emmerich and B. Naujoks. Metamodel-assisted multiobjective optimization with implicit constraints and its application in airfoil design. In *International Conference & Advanced Course ERCOFTAC, Athens, Greece*, 2004.
18. M. Emmerich, K. Yang, A. Deutz, H. Wang, and C.M. Fonseca. A multicriteria generalization of bayesian global optimization. *Advances in Stochastic and Deterministic Global Optimization*, pages 229–242, 2016.
19. B. Galván, D. Greiner, J. Periaux, M. Sefrioui, and G. Winter. Parallel evolutionary computation for solving complex cfd optimization problems: A review and some nozzle applications. *Parallel Computational Fluid Dynamics 2002*, pages 573–604, 2003.
20. K.C. Giannakoglou. Optimization and inverse design in aeronautics: how to couple genetic algorithms with radial basis function networks. *Innovative Tools for Scientific Computation in Aeronautical Engineering*, 2001.
21. K.C. Giannakoglou. Ingenet: A european thematic network for design optimization using computational intelligence. *Engineering Optimization*, 34(6):647–656, 2002.
22. A.P. Giotis and K.C. Giannakoglou. Evolutionary algorithms with surrogate models in multiobjective aerodynamic optimization. *Computer Methods in Applied Mechanics and Engineering*, 175(3):331–347, 1999.
23. A. Habib, H.K. Singh, T. Chugh, T. Ray, and K. Miettinen. A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 23(6):1000–1014, 2019.
24. I. Hupkens, A. Deutz, K. Yang, and M. Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 65–79. Springer, 2015.
25. S. Jeong and S. Obayashi. Efficient global optimization (ego) for multi-objective problem and data mining. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2138–2145. IEEE, 2005.
26. Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.
27. Y. Jin, M. Olhofer, and B. Sendhoff. On evolutionary optimization with approximate fitness functions. In *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pages 786–793, 2000.
28. D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
29. A.J. Keane. Statistical improvement criteria for surrogate-assisted multiobjective optimization. *AIAA Journal*, 38(2):211–219, 2000.
30. J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. In *IEEE Congress on Evolutionary Computation*, pages 325–332, 2006.
31. J. Knowles. Parego: A hybrid algorithm with on-line learning for multiobjective optimization problems. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization (EMO 2006)*, pages 394–409, Heidelberg, 2006. Springer.
32. D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52(6):119–139, 1951.

33. Ł. Łaniewski-Wołłk, S. Obayashi, and S. Jeong. Development of expected improvement for multi-objective problem. In *Proceedings of the 42nd Fluid Dynamics Conference/Aerospace Numerical Simulation Symposium*, 2010.
34. M. Laumanns and J. Ocenasek. Bayesian optimization algorithms for multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 298–307. Springer, 2002.
35. R.A.E. Mäkinen, J. Périaux, and J. Toivanen. Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms. *International Journal for Numerical Methods in Fluids*, 30(2):149–159, 1999.
36. G. Matheron. Principles of geostatistics. In *Economic Geology*, pages 1246–1266. Society of Economic Geologists, 1963.
37. J. Mockus. On bayesian methods for seeking the extremum. In *Proceedings of the IFIP Technical Conference*, pages 400–404, 1974.
38. J. Mockus, V. Tiesis, and A. Žilinskas. Bayesian methods for seeking the extremum. In *Toward global optimization*. Elsevier, 1978.
39. D.E. Myers. Kriging, cokriging, radial basis functions and the role of positive definiteness. *Computers & Mathematics with Applications*, 24(12):139–148, 1992.
40. W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted-metric selection. In *International Conference on Parallel Problem Solving from Nature*, pages 784–794. Springer, 2008.
41. J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.
42. C. M. Siefert. Model-assisted pattern search. Undergraduate Honors Thesis, College of William & Mary, Virginia, USA, 2000. Advisors: Virginia J. Torczon, Michael W. Trosset, and Stephen K. Park.
43. H. Ulmer, F. Streichert, and A. Zell. Model assisted evolution strategies. In *Knowledge Incorporation in Evolutionary Computation*, pages 333–355. Springer, 2005.
44. T. Wagner, M. Emmerich, A. Deutz, and W. Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *PPSN XI*, pages 718–727. Springer, 2010.
45. C. Williams and C. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8, 1995.
46. K. Yang, M. Emmerich, A. Deutz, and C.M. Fonseca. Computing 3-d expected hypervolume improvement and related integrals in asymptotically optimal time. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 685–700. Springer, 2017.
47. A. Žilinskas. On similiarities between two global optimization algorithms based on different (bayesian and lipschitzian) approaches. *Springer Optimization and Its Applications*, pages 233–237, 2025.
48. A. Žilinskas and J. Žilinskas. Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems. *Communications in Nonlinear Science and Numerical Simulation*, 21(1-3):89–98, 2015.
49. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.