Article

Gaussian Process Regression Based Multi-Objective Bayesian Optimization for Power System Design

Nicolai Palm ©, Markus Landerer and Herbert Palm ®

Systems Engineering Laboratory, University of Applied Sciences, Lothstrasse 64, 80335 München, Germany
* Correspondence: herbert.palm@hm.edu

Abstract: Within a disruptively changing environment, design of power systems becomes a complex task. Meeting multi-criteria requirements with increasing degrees of freedom in design and simultaneously decreasing technical expertise strengthens the need for multi-objective optimization (MOO) making use of algorithms and virtual prototyping. In this context, we present Gaussian Process Regression based Multi-Objective Bayesian Optimization (GPR-MOBO) with special emphasis on its profound theoretical background. A detailed mathematical framework is provided to derive a GPR-MOBO computer implementable algorithm. We quantify GPR-MOBO effectiveness and efficiency by hypervolume and the number of required computationally expensive simulations to identify Pareto-optimal design solutions, respectively. For validation purposes, we benchmark our GPR-MOBO implementation based on a mathematical test function with analytically known Pareto front and compare results to those of well-known algorithms NSGA-II and pure Latin Hyper Cube Sampling. To rule out effects of randomness, we include statistical evaluations. GPR-MOBO turns out as an effective and efficient approach with superior character versus state-of-the art approaches and increasing value-add when simulations are computationally expensive and the number of design degrees of freedom is high. Finally, we provide an example of GPR-MOBO based power system design and optimization that demonstrates both the methodology itself and its performance benefits.

Keywords: power system design; multi-objective optimization; gaussian process regression; Bayesian Optimization; expected hypervolume improvement; squared exponential kernel

1. Design of Complex Power Systems

Design of power systems is facing a triple disruptive upheaval: (i) primary energy sources are being converted from fossil to renewable, (ii) grid topologies are moving from centralized to decentralized dominance, and (iii) previously separate sectors (heat, electricity, and mobility) are merging. Driven by these changes, emerging technologies will replace established ones, while at the same time the number of degrees of freedom in facility design will increase. The resulting VUCA (volatile, uncertain, complex and ambiguous) [1] environment poses significant risks for all involved stakeholders throughout the value chain, from product owners over system architects to operators and consumers. Simultaneous ecologic and economic success of power plants requires the ability to identify system design alternatives with multi-objective optima in terms of cost-benefit trade-off for individual use cases.

The lack of technical experience around emerging or new technologies significantly increases the challenge of optimization. Virtual prototyping (VP) guided by engineering experience, therefore, may be considered the standard approach for today’s energy facility design. However, with an increasing number of technology alternatives and decreasing system-level technical experience, the dependence between objectives and design parameters is unknown in many cases. Power system design, thereby, becomes a black box multi-objective optimization (MOO) problem.
VP-based MOO guided by engineering intuition again fails when engineering experience is lacking and full or fractional factorial coverage of the design space is not a viable option due to (i) the dimensionality (i.e., number of degrees of freedom) of the design space and (ii) the time-consuming and costly effort required to simulate power systems. Therefore, gaining engineering knowledge about the relationship between system design objectives as a function of design parameters based on as few simulations as possible is of paramount importance.

VUCA driven power system design, accordingly, requires MOO algorithms being (i) effective, i.e., knowledgable of multi-objective (MO) related optimal cost-benefit (lateron called Pareto-optimal) trade-offs and (ii) efficient, i.e., requiring only a limited number of required (computationally expensive) simulations for quantifying these trade-offs.

While a broad variety of MOO approaches in black box environments deals with effective algorithms, only few of them meet the efficiency criterion [2]. Specifically Bayesian Optimization (BO) [3–6] algorithms based on Gaussian Process Regression (GPR) [3,7–9] appear as interesting effective and efficient MOO candidates.

GPR [10,11] provides a regression model that (in contrast to alternative methodical approaches) (a) does not suffer from the “curse of dimensionality” [12] ([Section 3.1]), and (b) inherently provides a quantification of regression uncertainty of the model. This uncertainty is exploited by the BO approach [13] which in turn is used to optimize the surrogate. For a more detailed and fundamental comparison of MOO approaches, we recommend [14].

In this paper, we present the mathematical background of GPR-based Bayesian Optimization (GPR-MOBO) in detail. This includes statements and selected proofs of key results. With that theoretical foundation on hand, we derive a computer implementation of an introduced GPR-MOBO algorithm, quantify its effectiveness and efficiency, and demonstrate the superiority of GPR-MOBO over state-of-the-art MOO algorithms, including a GPR-MOBO application to a power system design example. The paper is structured as follows: Section 2 restates the introductory problem in mathematical terms, including definitions such as “Pareto-optimality” or “hypervolume”. Section 3 explains Gaussian Process Regression (GPR) and a Bayesian Optimization (BO) based on it before a bipartite validation follows in Section 4: First, the superiority of the presented approach is validated via a mathematical test function before the proposed GPR-MOBO approach is applied to the design and optimization of a real life power system. Section 6 discusses our results, gives a brief summary and highlights ideas for future work.

2. Problem Statement in Mathematical Terms

Within this chapter, we phrase the task of effectively and efficiently identifying objective trade-offs in mathematical terms. For this purpose, we consider an (unknown) \( t \)-dimensional black box function

\[ f : X \rightarrow \mathbb{R}^t. \]

We refer to \( X \subset \mathbb{R}^d \) as the design space with target space \( \mathbb{R}^t \) for \( d, t \in \mathbb{N} \). Assume further, we can sample \( f \) at a finite number of points, i.e., choosing \( x_1, ..., x_N \) we obtain \( f(x_1), ..., f(x_N) \). We translate the MOO issue

\[ d_{opt} = \arg\min_{d \in X} f(d) \]

of identifying an optimal design \( d_{opt} \in X \) to finding solutions of \( f \) which represent non-dominated (Pareto-optimal) trade-offs of \( f \), i.e., we are looking for Pareto points of \( f \):
**Definition 1** (Pareto point and front). We write
\[
  t \preceq s \iff \forall i: t_i \leq s_i \\
  t \prec s \iff \forall i: t_i < s_i.
\]

Then, given a set \( S \subset \mathbb{R}^t \), a point \( s \in S \) is called Pareto point, if there exists no other point \( t \in S \) satisfying \( t \preceq s \) and \( t_i < s_i \) for some component \( i \). The set of all Pareto (optimal) points is called the Pareto front of \( S \).

More general, an \( x \in X \) is a Pareto point of \( f \) if \( f(x) \) is a Pareto point of \( f(X) = \{ f(x) : x \in X \} \). The set of all such points is called Pareto front of \( f \).

Based on this definition, we will call a MOO algorithm to be effective whenever it is capable to identify the set of non-dominated (Pareto-optimal) trade-offs for the (unknown) black box function. Introducing a measure of effectiveness, we will now define the so-called hypervolume.

**Definition 2** (Hypervolume). Denote by
\[
  I : \mathbb{R}^t \times \mathbb{R}^t \rightarrow \mathcal{P}(\mathbb{R}^t), \quad (x, r) \mapsto \{ z \in \mathbb{R}^t : x \preceq z \prec r \}
\]
the function sending two \( t \)-dimensional real vectors to the cube bounded by them where \( \mathcal{P} \) denotes the power set. The Hypervolume of some (finite) set \( Y \subset \mathbb{R}^t \) with respect to a reference point \( r \in \mathbb{R}^t \) is given by the Lebesgue-measure
\[
  \text{HV}_r(Y) = \lambda_t \left( \bigcup_{y \in Y} I(y, r) \right)
\]
of the union over all cubes bounded by the reference point and by some point in \( Y \).

**Figure 1** illustrates the Hypervolume on exemplary base in two dimensions.

![Figure 1](image.png)

**Figure 1.** Hypervolume (blue area) in \( \mathbb{R}^2 \). The dotted rectangles illustrate the volumes i.e., \( I(u, r) \) spanned by the respective point \( u \in U \) (\( \bullet \)) and the reference point \( r \) (\( \times \)).

The Hypervolume is closely related to Pareto points in the following sense.

**Proposition 1.** Let \( S \subset \mathbb{R}^t \) be quasi-compact (i.e., bounded and closed) and \( U \subset S \) be a finite subset. Let \( r \in \mathbb{R}^t \) such that
\[
  \text{HV}_r(U \cup \{ s \}) > \text{HV}_r(U)
\]
for some \( s \in S \). Then,
\[
  p' = \arg\max_{p \in S} \text{HV}_r(U \cup \{ p \})
\]
is a Pareto point of \( S \).

**Proof.** See Appendix A. \( \square \)
In simple words, Proposition 1 states that maximizing the hypervolume by adding an image (black box function) point implies this point to be a Pareto point. Accordingly, hypervolume is a suitable indicator of MOO effectiveness while the number of (simulation) samples required to find such Pareto points itself is a suitable measure for efficiency.

3. Bayesian Optimization Based on Gaussian Process Regression

Gaussian Process Regression (GPR) based Bayesian Optimization (BO) using Expected Hypervolume Improvement (EHVI, see Section 3.3) as acquisition function is a promising algorithmic approach to meet the simultaneous goals of effectiveness and efficiency. The following sub-sections will introduce the general mathematical GPR background (Section 3.1), choice of GPR related hyperparameters (Section 3.2) and GPR based multi-objective BO (GPR-MOBO, Section 3.3) prior summarizing previous sub-sections as a mathematical base for the subsequent algorithmic implementation.

3.1. Gaussian Process Regression

We summarize and recall the definition, statements and formulas needed in order to properly apply Gaussian Process Regression (GPR).

3.1.1. Multivariate Normal Distribution

Let \( n \in \mathbb{N} \) be a positive integer and \( C \in M_n(\mathbb{R}) \) be a real, positive definite matrix of dimension \( n \) with \( M_n(\mathbb{R}) \) being the space of \( n \times n \) matrices with values in \( \mathbb{R} \). Let \( m \in \mathbb{R}^n \) be an \( n \)-dimensional real vector. Recall the multivariate normal distribution \( \mathcal{N}(m, C) \) to be the probability measure on \( \mathbb{R}^n \) induced by the density function

\[
f_{C,m}: \mathbb{R}^n \rightarrow \mathbb{R}, \quad x \mapsto \frac{\exp\left(-0.5(x - m)^\top C^{-1}(x - m)\right)}{\sqrt{\det(C)(2\pi)^n}}.
\]

The vector \( m \) is called mean(-vector) and the matrix \( C \) is called covariance matrix of \( \mathcal{N}(m, C) \).

Multivariate normal distributions are stable under conditioning in the following sense.

**Theorem 2.** Let \( X_1, X_2 \) be two random variables such that \( (X_1, X_2) \) is multivariate normal \( \mathcal{N}(m, C) \)-distributed with mean \( m = (m_1, m_2) \in \mathbb{R}^p \times \mathbb{R}^q \) and covariance matrix

\[
C = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} \in M_{p+q}(\mathbb{R}), \quad C_1 \in M_p(\mathbb{R}).
\]

Then, given some \( x \in \mathbb{R}^q \) in the co-domain of \( X_2 \), the conditional density function of \( X_1 \) given \( X_2 = x \) is given by \( f_{C',m'} \) with

\[
m' = m_1 + C_2 C_4^{-1}(x - m_2)
\]

and

\[
C' = C_1 - C_2 C_4^{-1} C_3.
\]

**Proof.** Theorem 2.5.1 in [15]. \( \Box \)

In particular, conditioning a multivariate normal distribution turns out to be multivariate normal distributed as well.

3.1.2. Stochastic Processes

Let \( (\Omega, \mathcal{F}, P) \) be a probability space, \( I \) a set and \( (S, \mathcal{F}') \) be a measurable space.

**Definition 3.** A stochastic process with state space \( S \) and index set \( I \) is a collection of random variables \( X_i: \Omega \rightarrow S, i \in I \).
Remark 1. Recall that arbitrary products of measurable spaces exist and their underlying sets are given by the Cartesian products of sets. By the universal property of the product, a stochastic process \( X_i, i \in I \), therefore, consists of the same data as a measurable function \( X : \Omega \to S^I \).

Given a stochastic process \( X : \Omega \to S^I \), in practice, we are mostly interested in the induced measure \( P_X = P(X^{-1}(-)) \) on \( S^I \). On the other hand, given such probability measure \( P_I \) on \( S^I \), we obtain a stochastic process \( pr_U : S^I \to S \) given by the canonical projections. In that sense, a stochastic process may be seen as a proper construction of a probability measure on the product space \( S^I \).

3.1.3. Gaussian Process

With the definition of stochastic processes on hand, we can generalize the multivariate normal distribution (defined on finite products of real numbers) to possible infinite products of real numbers in the following sense.

**Definition 4 (Gaussian Process).** Let \( X \) be a set. A Gaussian Process with index set \( X \) is a family of real valued random variables \((r_x)_{x \in X}\) such that for every finite subset \( U \subseteq X \), the random variable \((r_u)_{u \in U}\) is multivariate normal distributed.

Recall that by the above, this induces a probability measure on \( \mathbb{R}^I \). We can “construct” Gaussian Processes in the following way:

**Theorem 3.** Let \( X \) be a set, \( C : X \times X \to \mathbb{R} \) be a positive quadratic form in the sense that for every finite subset \( U \subseteq X \) the induced matrix

\[
C_U = (C(u_1, u_2))_{u_1, u_2 \in U}
\]

is positive definite and \( m : X \to \mathbb{R} \) be a function. Given a subset \( U \subseteq X \), denote by

\[
m_U = (m(u))_{u \in U}
\]

the induced vector.

Then, there exists a unique probability measure \( P \) on \( \mathbb{R}^X \) satisfying

\[
P_{pr_U} = P \circ pr_U^{-1} = N(m_U, C_U)
\]

for all \( U \subseteq X \) finite where \( pr_U : \mathbb{R}^X \to \mathbb{R}^U \) denotes the canonical projection.

The function \( C \) is called covariance function and \( m \) is called mean function of \( P \).

**Proof.** See Appendix B. \( \square \)

In other words, we construct Gaussian Processes by choosing a positive quadratic form \( C \) further referred to as covariance function and a mean function \( m \).

**Example 4 (Squared exponential kernel).** The squared exponential kernel

\[
k_{l, \sigma^2} : X \times X \to \mathbb{R}, \quad (x, y) \mapsto \sigma^2 \exp \left( \frac{1}{2l^2} \| x - y \|^2 \right)
\]

(1)

is a covariance function (i.e., a positive quadratic form; see [16]) for every \( l, \sigma^2 > 0 \). The parameter \( l \) is called lengthscale and the parameter \( \sigma^2 \) is called output variance. Other covariance functions may also be found in [16].
Example 5 (Covariance with white Gaussian noise). Let \( m \) be a function and \( C \) be a covariance function. Given \( U = \{x_1, ..., x_n\} \subset X \). The reader may convince himself that the function

\[
C_{\text{wgn}} : X \times X \to \mathbb{R}, (x, y) \mapsto \begin{cases} C(x, x) + \sigma^2 & x = y \in U \\ C(x, y) & \text{else} \end{cases}
\]

is a positive quadratic form for each \( \sigma^2 > 0 \). Note that \( \sigma^2 \) may be considered as a hyperparameter.

Combining Theorems 2 and 3, we derive an appropriate “conditioned” Gaussian Process.

Corollary 6. Let \((r_x)_{x \in X}\) be a Gaussian Process with index set \(X\),

\[
C : X \times X \to \mathbb{R}
\]

its covariance function and

\[
m : X \to \mathbb{R}
\]

its mean function. Let \( U \subset X \) be a finite subset consisting of \( n \in \mathbb{N} \) elements and \( y \in \mathbb{R}^n \).

Then, there exists a unique probability measure \( P \) on \( \mathbb{R}^X \) such that for every finite subset \( J \subset X - U \) the density function of \( P \circ pr_J^{-1} \) is given by the conditional density function of \((r_j)_{j \in J}, (r_u)_{u \in U}\) given \( (r_u)_{u \in U} = y \).

Its mean function \( m' \) and covariance function \( C' \) are constructed as follows: For every \( x \in X \) define

\[
C_{x,U} = (C(x, u))_{u \in U}.
\]

Then,

\[
m'(x) = m(x) + C_{x,U}^T C_U^{-1} (y - m_U) \tag{2}
\]

and

\[
C'(x_1, x_2) = C(x_1, x_2) + C_{x_1,U}^T C_U^{-1} C_{x_2,U}. \tag{3}
\]

Proof. See Appendix B. \( \square \)

3.1.4. Gaussian Process Regression

Consider a supervised learning problem

\[
f : X \to \mathbb{R}
\]

with training points

\[
T = \{(x_1, y_1), ..., (x_N, y_N)\} \subset X \times \mathbb{R}.
\]

The task is to find an appropriate approximation of the unknown (“black box”) function \( f \). To solve this task, we may use Gaussian Process regression, the idea of which is to

(i) define a Gaussian Process on \( X \) by defining a mean and covariance function on \( X \) (Theorem 3),
(ii) condition that Gaussian Process in the sense of Corollary 6 with \( U = \{x_1, ..., x_N\} \) and \( y = (y_1, ..., y_N) \) and
(iii) use \( m' \) from Formula 2 as approximation of \( f \).

A GPR for \( f \) and \( T \) is then the data of a Gaussian Process on \( X \) conditioned to \( U = \{x_1, ..., x_N\} \) and \( y = (y_1, ..., y_N) \).
Remark 2. By its very nature, a GPR is equipped with a natural measure of prediction uncertainty. Instead of a single point prediction \( y \) for \( f(x) \) with \( x \in X \), we obtain a probability distribution \( N(m'(x), C'(x,x)) \).

We interpret \( N(0, C'(x,x)) \) as the uncertainty in the prediction \( m'(x) \) at \( x \).

Figure 2 illustrates the conditioning of a GPR to some new evaluations.

Figure 2. Top: real function \( f = \sin (\cdot \cdot \cdot) \) and GPR with mean \( m (\cdot \cdot \cdot) \) and covariance function \( C \) where (\( \cdot \cdot \cdot \)) symbolizes mean plus/minus standard deviation (i.e., \( m(x) \pm \sqrt{C(x,x)} \)) at each point. Bottom: Same as top with \( m \) and \( C \) conditioned to the vertical cyan dashes.

3.2. GPR Hyperparameter Adaption

Using a GPR for supervised learning problems requires the choice of some (initial) mean function and covariance function (Theorem 3). Most examples of covariance functions involve the choice of hyperparameters. Example 4 involves the choice of a lengthscale and output variance.

Consider a supervised learning problem \( f: X \rightarrow \mathbb{R} \) with training points \( T = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset X \times \mathbb{R} \). Given a mean function \( m \) and a family of covariance functions \( C_\theta \) with \( \theta \) element of some index set \( \vartheta \), we choose a hyperparameter \( \theta \) by following the maximum likelihood principle.

Denote by \( f_{C,m,\theta} \) the density function of the multivariate normal distribution \( N(\{x_1, \ldots, x_N\}, C_{\vartheta_{\{x_1, \ldots, x_N\}}} \).

We choose \( \theta \) by solving

\[
\theta^* = \arg\max_{\theta \in \vartheta} (f_{C,m,\theta}(y_i)_{i=1,\ldots,N}).
\] (4)

Remark 3. In practice, one often replaces \( f_{C,m,\theta} \) with \( \ln f_{C,m,\theta} \) and solves

\[
\theta^* = \arg\max_{\theta \in \vartheta} (\ln f_{C,m,\theta}(y_i)_{i=1,\ldots,N})
\] (5)

resulting in identical parameters. However,

\[
\ln f_{C,m,\theta}(y) = -\frac{1}{2} y^\top C_{\vartheta_{\{x_1, \ldots, x_N\}}} y - \frac{1}{2} \ln(\det(C_{\vartheta_{\{x_1, \ldots, x_N\}}})) - \frac{N}{2} \ln(2\pi)
\]

is more convenient to work with.
3.3. Bayesian Optimization

We define the \textit{hypervolume improvement} as the gain of hypervolume when adding new points. At some places, the underlying function used for calculating new sample (or \textit{infill}) points is called \textit{acquisition function}.

**Definition 5** (Hypervolume Improvement). Given a reference point \( r \in \mathbb{R}^t \) and a finite set of vectors \( F \subset \mathbb{R}^t \), the hypervolume improvement of some \( y \in \mathbb{R}^t \) is defined as

\[
HVI_r(F, y) = HV_r(F \cup \{ y \}) - HV_r(F). \tag{6}
\]

We denote by

\[
HVI_{r,F} : \mathbb{R}^t \to \mathbb{R}, y \mapsto HVI_r(F, y) \tag{7}
\]

the resulting function. We often write \( HVI_r \) instead of \( HVI_{r,F} \) whenever \( F \) is clear from context.

Observe that \( HVI_{r,F} \) is continuous (see Appendix C), hence integrable, on a bounded subset.

**Remark 4.** Maximizing the Hypervolume improvement results in Pareto points (see Proposition 1).

Consider a black box function \( f : X \to \mathbb{R}^t \) with evaluations \( T = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset X \times \mathbb{R}^t \). Given an approximation \( \hat{f} \) of \( f \) (such as the mean of a GPR) and a suitable reference point \( r \in \mathbb{R}^t \), we strive to calculate

\[
\arg \max_{x \in X} HVI_r(\{ y_1, \ldots, y_N \}, \hat{f}(x))
\]

in order to find a preimage of a Pareto point of \( f \).

Recall that GPRs include a prediction uncertainty measure (Remark 2). We can take this additional information into account when maximizing the Hypervolume improvement in the following ways.

**Definition 6** (Expected Hypervolume Improvement). Let mean functions \( m_i : X \to \mathbb{R} \) and covariance functions \( C_i : X \times X \to \mathbb{R} \) on \( X \) for \( i = 1, \ldots, t \) be given. Denote by

\[
m(x) = (m_i(x))_{i=1,\ldots,t}
\]

the induced mean vector and by \( C(x) \) the diagonal matrix with

\[
C(x)_{ii} = C_i(x,x).
\]

Then, the expected Hypervolume improvement at \( x \in X \) is given by the expected value

\[
EHVI(x) = \int_{y \in \mathbb{R}^t} HVI_r(y) dP_x = \int_{y \in \mathbb{R}^t} HVI_r(y, f_m(x), C(x)) dy \tag{8}
\]

of \( HVI \) with respect to the probability measure

\[
P_x = \otimes_{i=1}^t \mathcal{N}(m_i(x), C_i(x,x)) = \mathcal{N}(m(x), C(x))
\]

on \( \mathbb{R}^t \).

In many situations, the training data (more precisely the evaluations) are manipulated (i.e., \textit{pre-processed}) before training (i.e., calculating the hyperparameters of the means und covariance functions). By the very definition, we obtain the following corollary:
Corollary 7. Let \( S \subset \mathbb{R}^t \) and \( g : S \to \mathbb{R}^t \) be a function. Assume \( g \) satisfies \( a_i < b_i \) if and only if \( g(a)_i < g(b)_i \) for all \( i = 1, \ldots, t \) and \( a, b \in S \). Then, \( s \in S \) is a Pareto point if and only if \( g(s) \in g(S) \) is a Pareto point.

Therefore, any function satisfying the above assumptions may be used for pre-processing of data points in the context of multicriterial optimization.

Remark 5. The expected Hypervolume improvement involves the choice of some reference point. By construction, this choice affects the (expected) hypervolume contribution of any point in the target space. Notice that the reference point must be strictly greater in every component than every pareto optimal solution in order to ensure a hypervolume greater than zero for every such point. For example, if the black box function \( f \) factorizes through \([0,1]^t \subset \mathbb{R}^t\), then, such a reference point may be chosen by

\[
(1 + \varepsilon, \ldots, 1 + \varepsilon) \in \mathbb{R}^t
\]

for some \( \varepsilon > 0 \).

Further discussion for the reference point selection may be found in the literature, e.g., under [17].

Roughly speaking, the expected hypervolume improvement is an extension of the hypervolume improvement incorporating the uncertainty information encapsulated in the GPRs. One hopes that maximizing the expected hypervolume improvement (of the GPRs) maximizes the hypervolume improvement of the black box function more efficiently than simply maximizing the hypervolume of the underlying mean function of the GPRs. Note that incorporating the uncertainty (of the model) allows to reflect a trade-off between exploration and exploitation.

3.4. Summary—Base for an Algorithmic Implementation

We shortly summarize the necessary steps in order to apply GPR and EHVI based multicriterial optimization to a black box function \( f : X \to \mathbb{R}^t \).

Let

\[
T = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subset X \times \mathbb{R}^t
\]

be evaluations of \( f \). We define \( X_0 = \{x_1, \ldots, x_N\} \) and \( Y_0 = \{y_1, \ldots, y_N\} \).

3.4.1. Setting Up the GPRs

For each \( i = 1, \ldots, t \) we choose a mean function \( m_i \) and a covariance function (i.e., a positive quadratic form) \( C_i \) on \( X \). Examples of covariance functions can be found in [16]. By Theorem 3, we obtain a Gaussian Process for each \( i \). In case the covariance function involves the choice of some hyperparameter, we determine that parameter by solving Equation (5). Next, we condition each mean \( m_i \) and covariance function \( C_i \) to \( T \) using Equations (2) and (3), respectively. We obtain GPRs, defined by their conditioned mean \( m'_i \) and covariance function \( C'_i \) for each \( i = 1, \ldots, t \) (i.e., for each output component).

3.4.2. Maximizing the EHVI

We maximize the expected Hypervolume improvement, i.e., we solve

\[
p = \arg\max_{x \in X} \text{EHVI}(x)
\]

according to Equation (8) with respect to the mean functions \( m'_i \) and covariance functions \( C'_i \). Algorithms for calculation of the expected Hypervolume improvement may be found in [18,19]. Lastly, we evaluate the black box function \( f \) at the found \( p \).

We close this section by a couple of practical remarks:
• GPRs form a rich class of regression models. However, evaluating a GPR involves the inversion of a $N \times N$ matrix with $N$ being the number of training points (see Equation (3)). Accordingly, evaluating a GPR tends to get slow with increasing number of training points.

• In addition, GPRs (as any regression model) requires a careful pre-processing of the data in order to produce reasonable results. At the very least, the input and output of the training data should be normalized (e.g., to $(0, 1)$).

To enable the reader understanding the GPR-MOBO algorithm described below in a comprehensible way, we present here a pre-processing example of the training data: Denote by $x_{\min} = (\inf_{z \in X}(z_i))_{i=1,...,d}$ and $x_{\max} = (\sup_{z \in X}(x_i))_{i=1,...,d}$ the componentwise minimum resp. maximum within the design space $X$. Then, define

$$n_X: X \rightarrow [0, 1]^d, \quad x \mapsto \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

(11)

where the division is performed componentwise and $X = n_X(X_0)$. Assuming $X$ to be bounded and $x_{\max,i} \neq x_{\min,i}$ for all components $i$, this is well defined. Furthermore, we define $y_{\min} = (\min_{y \in Y_0}(y_i))_{i=1,...,t}$ and $y_{\max} = (\max_{y \in Y_0}(y_i))_{i=1,...,t}$ by componentwise minimum resp. maximum of the evaluations. Without loss of generality, we may assume $y_{\max,i} \neq y_{\min,i}$ for all $i$. Define

$$n_Y: \mathbb{R}^t \rightarrow \mathbb{R}^t, \quad y \mapsto \frac{y - y_{\min}}{y_{\max} - y_{\min}}$$

(12)

and $Y = n_Y(Y_0)$. It is straightforward to check that $n_Y$ satisfies the assumptions of Corollary 7.

We obtain normalized training data

$$T_n = \{(n_X(x_1), n_Y(y_1)), \ldots, (n_X(x_N), n_Y(y_N))\} \subset [0, 1]^d \times [0, 1]^t.$$ 

4. Algorithmic Implementation and Validation

In this section, we first derive an algorithmic implementation of an adaptive Bayesian Optimization (BO) algorithm based on GPR infill making use of expected hypervolume improvement as acquisition function. In Section 4.2, we apply this algorithm to a mathematical test function with known Pareto front prior validating it in Section 4.3 within context of a real-world power system application.

4.1. GPR-MOBO Algorithmic Implementation

The proposed GPR-MOBO workflow presented in Algorithm 1 strictly follows the sequence of steps for applying GPR with EHVI-based MOO to a black box function as described in Section 3.4.

Wherever applicable, we refer to equations as referenced in Section 3. As an initial design of experiments (DoE) $X_0$, we propose Latin Hypercube Sampling (LHCS) according [20] as the basis of the initial computationally expensive black box function evaluation $f(X_0)$. We assume the initial sampling by $N_0$ expensive evaluations to cover the full range of parameter values within the target space to guarantee its image on unit cube after normalization.

$N_{\text{tot}}$ (maximum number of samples available for computationally expensive black box function evaluations) and $N_0$ (fraction of $N_{\text{tot}}$ as the number of initial samples) may be considered as the GPR-MOBO hyperparameters.
Algorithm 1 Structural MO Bayesian Optimization workflow.

```latex
\begin{align*}
X_0 &\leftarrow \{x_1, \ldots, x_{N_0}\} & \text{\(\triangleright\) Set initial DoE} \\
Y_0 &\leftarrow f(X_0) & \text{\(\triangleright\) Simulate at } X_0 \text{ (expensive)} \\
m &\leftarrow 0_{\text{map}} & \text{\(\triangleright\) Choose zero map as mean function} \\
C &\leftarrow (1) & \text{\(\triangleright\) Choose covariance function} \\
r &\leftarrow (9) & \text{\(\triangleright\) Choose reference point} \\
\text{while} |X_0| < N_{\text{tot}} \text{ do} & \text{\(\triangleright\) unless abortion criterion is met} \\
X_n, Y_n &\leftarrow (11), (12) & \text{\(\triangleright\) Pre-process data} \\
\text{for } i = 1 \ldots t \text{ do} & \text{\(\triangleright\) for each dimension of target space} \\
\theta_i^* &\leftarrow (5) & \text{\(\triangleright\) Calculate hyperparameters} \\
m_i^* &\leftarrow (2) & \text{\(\triangleright\) Condition mean} \\
C_i^* &\leftarrow (3) & \text{\(\triangleright\) Condition covariance} \\
\text{end for} & \text{\(\triangleright\) Calculate optimal infill sample point} \\
p_0 &\leftarrow (10) & \text{\(\triangleright\) Add infill to } X_0 \\
X_0 &\leftarrow X_0 \cup \{y^{-1}(p)\} & \text{\(\triangleright\) Add simulation at } y^{-1}(p) \text{ to } X_0 \text{ (expensive)} \\
Y_0 &\leftarrow Y_0 \cup \{f(y^{-1}(p))\} & \text{\(\triangleright\) Acquire according design space points} \\
\text{end while} \\
Y_p &\leftarrow \{y_1, \ldots, y_k\} \\
X_p &\leftarrow f^{-1}(Y_p) \\
\text{return } X_p, Y_p
\end{align*}
```

Algorithm 1 has been implemented in Matlab 2021b making use of the minimizer \texttt{fmincon} based on the \texttt{sqp} option (called multiple times with different initial values by \texttt{GlobalSearch}) to find the minimum of the negative log marginal likelihood function for hyperparameter adaption (see Section 3.2) and for BO (see Section 3.3) with application to the negative EHVI function which is provided by [21]. At this point it seems worthwhile noting, \texttt{GlobalSearch} not to yield deterministic results, i.e., multiple runs with identical input values may vary in their output values.

To make the computation more efficient, for numerical inversion of \(C_f\), the Cholesky decomposition is used.

4.2. Test Function Based Validation

In this subsection, we are aiming for an effectiveness and efficiency comparison of Algorithm 1 versus state-of-the-art alternatives LHCS [20] and NSGA-II [22]. We present results of their application to a well defined black box function with analytically well known Pareto front. In our case, we picked the to be minimized test function ZDT1 according [22]

\[
f : [0, 1]^d \rightarrow \mathbb{R}^2, \quad \left( \begin{array}{c} x_1 \\
\vdots \\
x_d \end{array} \right) \mapsto \left( x_1 \left( 1 - \sqrt{\frac{x_1}{g(x)}} \right) \right) \quad \text{with} \quad g(x) = 1 + \frac{9}{d-1} \sum_{i=2}^{d} x_i
\]

and \(2 \leq d \leq 30\) indicating the design space dimension. Note that ZDT1 exhibits a convex Pareto front independent of \(d\).

We compare results for the alternative Pareto front search algorithms granting \(N_{\text{tot}} = 50\) black box evaluations plus adding an NSGA-II analysis with \(N_{\text{tot}} = 200\) black box evaluations. For LHCS, all samples were spent within the initial run while Algorithm 1 started with \(N_0 = 30\) initial LHCS samples. In case of \(N_{\text{tot}} = 50\) total samples, the NSGA-II algorithm was applied in \(n_{\text{gen}} = 5\) generations with a population of \(n_{\text{pop}} = 10\) samples while in case of \(N_{\text{tot}} = 200\) total samples, \(n_{\text{gen}} = 8\) generations with a population of \(n_{\text{pop}} = 25\) samples were run. For statistical purposes, the algorithm evaluations were repeated fifty times with different random starting values. Design space dimension was chosen by \(d \in \{2, 3, 5, 10, 15, 20, 25, 30\}\). The reference point during optimization was chosen according Equation (9) with \(\epsilon = 2\%\). Making use of knowledge about the ZDT1
target domain, we fixed $y_{min} = (0, 0)$ for pre-processing. To quantify the search algorithms’ performance, the hypervolume with respect to reference point $(2, 10)$ in relation to the (known) maximal hypervolume is evaluated. The results are plotted in Figure 3.

Box plots in Figure 3 indicate statistical evaluations of the repeated search runs. GPR-MOBO results are drawn in red, LHCS results in blue, NSGA-II results for $N_{tot} = 50$ in brown and for $N_{tot} = 200$ in green.

![Figure 3](image_url)

**Figure 3.** Statistical evaluation of relative hyper volume from 50 repeated runs comparing performance of three MOO algorithms for a selected set of ZDT1 design space dimension $d$. Top three evaluations for GPR-MOBO ($N_0 = 30$ initial samples, red), LHCS (blue) and NSGA-II (with $n_{pop} = 10$ samples in $n_{gen} = 5$ generations, brown) are based on $N_{tot} = 50$ samples. Bottom figure evaluation for NSGA-II ($n_{pop} = 25$ samples in $n_{gen} = 8$ generations, green) is based on $N_{tot} = 200$ samples.

4.3. Power System Design Based Validation

Within the last subsection of this chapter, we apply the GPR-MOBO Algorithm 1 to the design and optimization of a power system example. Figure 4 illustrates both, the toolchain and the generic power system model as used for GPR-MOBO power system design and optimization validation.

For the energy domain specific “Power System Modeling and Simulation Environment” of the tool chain in our example, we use the commercial tool PSS®DE [23]. Connected to the “Script Control” (implemented in Python code) through an “xml interface”, adjustable design space parameters of the power system model receive dedicated parameter values as computed within the “Algorithmic Workflow” (execution of the GPR-MOBO algorithm in Matlab 2021b code). Data stored within the “Data Base” are accessible for “Statistical Evaluation” and “Visualisation” (all implemented in Python code). The generic power system model for our validation example is defined by a star topology connecting a standard $H_0$-profile [24] electric load with 20.2 GWh annual demand to three aggregate components (“Wind turbine”, “PV”, “Battery”) and a “Public Grid”. The “Power System Modeling and Simulation Environment” is simulating power system results in terms of well defined key performance indicators (KPI), CAPEX (capital expenditures for installation of the according power system), and CO$_2$ (amount of carbon dioxide for a given configuration to provide the total amount of energy). The KPI behavior of the “Power System Modeling and Simulation Environment” can therefore be viewed as a black box function.
Parameter value ranges defining the design space for the system of interest are listed in Table 1.

Table 1. Design space limits for the experiment.

| Design Parameter                     | Min | Max |
|--------------------------------------|-----|-----|
| PV peak power / MW<sub>p</sub>       | 0   | 41  |
| Wind turbine peak power / MW<sub>p</sub> | 0   | 17.1|
| Battery rack capacity / MAh          | 0   | 1.21|

As mentioned in Section 3.3, for the GPR-MOBO, the design and target space samples (design parameters and KPIs, respectively) are normalized according Equations (11) and (12).

The results for the experiment are shown in Figure 5. \(N_0 = 30\) initial samples and \(N_{\text{tot}} = 50\) samples in total were chosen as GPR-MOBO hyperparameters, while total emission of carbon dioxide (CO<sub>2</sub> in kilotons) and capital expenditures for acquisition and installation of aggregate components (CAPEX in million euros) were selected as to be evaluated trade-off KPIs. Making use of knowledge about the according target domain, we fixed \(y_{\min} = (0, 0)\) for pre-processing. The reference point during optimization was chosen according Equation (9) with \(\varepsilon = 2\%\).

Figure 5 shows the acquired subspace of the target space by the experiment. All sample points are marked by crosses, while the Pareto optimal solutions forming the Pareto front are highlighted by red crosses \(\times\) and the remaining portion of initial (30 LHCS) results are marked by cyan crosses \(\times\). In addition, an approximate Pareto front is plotted, resulting from a full-factorial design space latticing based on 1200 sample points. Non-dominated Pareto points are indicated by blue dots with a Pareto front completed by linear interpolation (marked by a blue dotted line \(\cdot\cdot\cdot\)).
5. Discussion

According Section 2, we may interpret the identified hypervolume of a black box as an indicator of effectiveness for a given Pareto front search algorithm. Putting that identified hypervolume in ratio to the number of computationally expensive evaluations required to identify this hypervolume, in turn, may be considered a suitable indicator of the algorithm’s efficiency. Applying these definitions to results obtained in the previous Section 4, Figure 3 clearly indicates:

(i) Given the limited low number of \( N_{\text{tot}} = 50 \) samples, all selected algorithms indicate a decreasing effectiveness over an increasing number of design space dimensions (i.e., degrees of design freedom, DoF).

(ii) GPR-MOBO outperforms the effectiveness of the other algorithms for all DoF \( d \), showing even for \( d = 30 \) (with \( N_{\text{tot}} = 50 \)) at the median more than 97% effectiveness (versus less than 71% for the other algorithms).

(iii) For \( N_{\text{tot}} = 50 \), the effectiveness of NSGA-II and LHCS appears comparable within statistical significance for all evaluated DoF \( n \).

(iv) For \( N_{\text{tot}} = 50 \) samples, GPR-MOBO outperforms NSGA-II even when applied with \( N_{\text{tot}} = 200 \) samples.

(v) Efficiency of GPR-MOBO significantly increases with increasing DoF \( n \) when compared to other algorithms.

On the other hand, it is worthwhile mentioning GPR-MOBO algorithm to require more computation time than standard LHCS or NSGA-II algorithms. Depending on the test function dimension \( d \) and hardware environment, GPR-MOBO runs (i.e., 20 iterations) took between 3 min and 11 min wherein the fraction for evaluating the black box function (ZDT1) can be treated as negligible while NSGA-II or LHCS runs took only less than a tenth portion of this time. This indicates that the GPR-MOBO is advisable, if the black box function is expensive to evaluate in terms of capital, time, or other resources.

The experimental validation using an unknown black box function whose result is shown in Figure 5 again validates the effectiveness and efficiency of the GPR-MOBO. The Pareto front as identified by \( N_{\text{tot}} = 50 \) black box evaluations is already very close to the front as approximated by \( N_{\text{tot}} = 1200 \) full-factorial lattice based black box evaluations. Some GPR-MOBO Pareto points are even dominating that identified by full-factorial lattice. The example, thereby, demonstrates effectiveness and efficiency of GPR-MOBO based power system design and optimization.

The results shown indicate a general superiority of GPR-MOBO over state-of-the-art algorithms. However, this has been demonstrated only on exemplary base. We therefore
point out the inadmissibility of a generalization of superiority. A general superiority of GPR-MOBO cannot and should not be derived from single, individual test functions or application examples. Such a fundamental superiority would have to be mathematically proven and would presumably require fundamental knowledge of the black box function itself or the Pareto front spanned by this black box function.

6. Conclusions and Outlook

In this paper, we tackled the challenge of power system design and optimization in VUCA environment. We proposed a Multi-Objective Bayesian Optimization based on Gaussian Process Regression (GPR-MOBO) in the context of power system virtual prototyping. After a mathematical reformulation of the challenge, we presented the background of Gaussian Process Regression, including hyperparameter adaption and use in the context of a Bayesian Optimization approach, focusing on the expected improvement in hypervolume. For validation purposes, we benchmarked our GPR-MOBO implementation statistically based on a mathematical test function analytically known Pareto front and compared results to those of well-known algorithms NSGA-II and pure Latin Hyper Cube Sampling. We demonstrated superiority of the GPR-MOBO approach over the compared algorithms, especially for high dimensional design spaces. Finally, we applied the GPR-MOBO algorithm for planning and optimization of a power system (energy park) in terms of selected performance indicators of exemplary character.

Concluding, GPR-MOBO turned out as an effective and efficient approach for power system design and optimization in VUCA environment with superior character when simulations are computationally expensive and the number of design degrees of freedom is high.

Some topics remain open for future investigation. Besides a performance comparison with other than already selected algorithms, some detailed questions within the GPR-MOBO family are worthwhile to be considered. This includes topics such as the choice of acquisition function, pre-processing, selection of the reference points when (expected) hypervolume (improvement) is put in focus and application of various global optimizers to name just a few. One level above, more not yet satisfactorily answered question address the extension to mixed-integer design spaces and issues related to constraint handling.

Author Contributions: Conceptualization, H.P.; methodology, H.P.; software, N.P. and M.L.; validation, N.P. and M.L.; writing—original draft preparation, H.P.; writing—review and editing, N.P., M.L. and H.P.; visualization, M.L. and H.P.; supervision, H.P.; project administration, H.P.; mathematical theory, N.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Siemens AG under the “Pareto optimal design of Decentralized Energy Systems (ProDES)” project framework.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Siemens AG for providing a free PSS®DE license for the power system design based validation part.

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

Appendix A. Maximization of Hypervolume and Pareto Points

Lemma A1. Let \( x, x', r \in \mathbb{R}^d \) with \( x \prec r \). Define

\[
e_i := (r_1, \ldots, r_{i-1}, \max(x_i, x'_i), r_{i+1}, \ldots, r_t)
\]
as vector \( r \) with \( i \)-th component replaced with the maximum of \( x_i \) and \( x_i' \). Then,

\[
I(x, r) = (I(x, r) \cap I(x', r)) \cup \bigcup_{i=1}^{t} I(x, c_i).
\]

**Proof.** First, we prove “\( \supseteq \)”:

Since \( c_i \nleq r \) for all \( i \), we obtain \( I(x, c_i) \subset I(x, r) \), hence, “\( \supseteq \)”.

Secondly, we prove “\( \subseteq \)”:

Given some \( z \in I(x, r) - I(x', r) \) there exists an \( i \) with \( x_i \leq z_i < x_i' \). Since \( z \prec r \), we obtain \( z \in I(x, c_i) \). This proves “\( \subseteq \)”.

Moreover, if \( \max(x_i, x_i') = x_i \), then, \( I(x, c_i) = \emptyset \), hence, \( z \in I(x, c_i) \) implies \( x_i \leq z_i < x_i' \) and thus \( z \not\in I(x', r) \). This proves that \( I(x, r) \cap I(x', r) \) and \( \bigcup_{i=1}^{t} I(x, c_i) \) are disjoint. \( \square \)

**Corollary A2.** For \( x, x', r \in \mathbb{R}^t \) with \( x \prec r \) and \( x_i < x_i' \), there exists \( x \prec d \nleq r \) with

\[
I(x, d) \subseteq I(x, r) - I(x', r).
\]

**Proof.** We may assume \( x' \prec r \), since otherwise \( I(x', r) = \emptyset \) and the claim follows for \( d = r \). Using Lemma A1, we obtain

\[
I(x, r) - I(x', r) = \bigcup_{j=1}^{t} I(x, c_j) \supseteq I(x, c_i).
\]

Furthermore, \( x \prec c_i \nleq r \) since \( x \prec r \) and \( x_i < x_i' \). Thus, \( d = c_i \) satisfies the claim. \( \square \)

**Corollary A3.** Let \( S \subset \mathbb{R}^t \) be a finite subset and \( x, r \in \mathbb{R}^t \) with \( x \prec r \). Assume further for every \( s \in S \) there exists \( i_s \) such that \( x_{i_s} \prec s_{i_s} \). Then, there exists \( c \in \mathbb{R}^t \) with \( x \prec c \nleq r \) and

\[
I(x, c) \subseteq I(x, r) - \bigcup_{s \in S} I(s, r).
\]

In particular, the Lebesgue measure \( \lambda(I(x, c)) > 0 \) of \( I(x, c) \) is greater than zero.

**Proof.** We prove the claim by induction over the size \( n \) of \( S \). For \( n = 1 \), this is the Corollary A2. Assume the claim holds for all such \( S \) with size equal to \( n \). Given \( S \) with cardinality \( n + 1 \), there exists such \( c \) with

\[
I(x, c) \subset I(x, r) - \bigcup_{\substack{s \in S, \ s \neq s_1}} I(s, r).
\]

Furthermore, we obtain

\[
I(x, r) - \bigcup_{s \in S} I(s, r) \supseteq I(x, c) - I(s_1, c) \supseteq I(x, c')
\]

for some \( x \prec c' \nleq c \) where the last subset is obtained by applying Corollary A2 to \( r = c, x' = s_1 \).

Lastly, the Lebesgue measure \( \lambda(I(x, c')) \) is given by the product \( \prod_{i=1}^{t} (c'_i - x_i) > 0 \) which is greater than zero since \( x \prec c' \). \( \square \)

**Corollary A4.** Let \( S \subset \mathbb{R}^t \) be a finite subset and \( x, r \in \mathbb{R}^t \) with \( x \prec s \nleq r \) for all \( s \in S \). Then,

\[
\text{HVL}_r(S, x) = 0
\]

if and only if there exists \( s \in S \) with \( s \nleq x \).
This proves the claim. Due to the additivity of the Lebesgue measure, it suffices to prove
\[ \lambda(I(x, r) - \bigcup_{s \in S} I(s, r)) > 0. \]

This follows by the Corollary A3.

**Corollary A5.** Let \( S \subseteq \mathbb{R}^l \) be a finite subset and \( x, x', r \in \mathbb{R}^l \) with \( x' < r \) for all \( s \in S \), \( x \preceq x' \) and \( x_i < x'_i \) for some \( i \). Assume further for every \( s \in S \) there exists \( i_s \) such that \( x_{i_s} < s_i \). Then,
\[ \text{HV}_r(S \cup \{x\}) > \text{HV}_r(S \cup \{x'\}) \geq \text{HV}_r(S). \]

**Proof.** Clearly, the last inequality holds. Due to
\[ \bigcup_{s \in S} I(s, r) \cup I(x, r) = \bigcup_{s \in S} I(s, r) \cup (I(x, r) - \bigcup_{s \in S} I(s, r)) \]
(resp. for \( x' \)) and the additivity of the Lebesgue measure, it suffices to prove
\[ \lambda(I(x, r) - \bigcup_{s \in S} I(s, r)) > \lambda(I(x', r) - \bigcup_{s \in S} I(s, r)). \]

By Corollary A3, there exists \( x \prec c \preceq r \) such that
\[ I(x, c) \subseteq I(x, r) - \bigcup_{s \in S} I(s, r). \]

Observe
\[ I(x, c) - I(x', r) = I(x, c) - I(x', c) \]
since \( c \preceq r \). Applying Corollary A2, we obtain some \( x \prec c' \preceq c \) such that
\[ I(x, c') \subseteq I(x, c) - I(x', r) = I(x, c) - I(x', c). \]

Observe
\[ I(x', r) \subseteq I(x, r) \]
since \( x \preceq x' \). Together we obtain,
\[ (I(x', r) - \bigcup_{s \in S} I(s, r)) \cup I(x, c') \subseteq I(x, r) - \bigcup_{s \in S} I(s, r). \]

By the additivity of the Lebesgue measure, we obtain
\[ \lambda(I(x, r) - \bigcup_{s \in S} I(s, r)) \geq \lambda(I(x', r) - \bigcup_{s \in S} I(s, r)) + \lambda(I(x, c')). \]

Since \( x \prec c' \), we argue as in the proof of the previous Corollary that \( \lambda(I(x, c')) > 0 \). This proves the claim.

**Proof of proposition 1.** We may without loss of generality assume \( s \prec r \) for all \( s \in S \).
Indeed, let \( T \subseteq S \) be the points in \( S \) satisfying \( s \prec r \) for all \( s \in T \). Given some pareto point \( t \in T \), assume there exists some \( s \in S - T \) with \( s \preceq t \). Then, \( s \preceq t \prec r \) which contradicts \( s \in S - T \).
Since $S$ is bounded and closed and $HVI_{r}(F, -)$ is continuous, we deduce the existence of some $p \in S$ which maximizes $HVI_{r}(F, -)$. Since

$$HVI_{r}(F, p) = \max_{x \in S} HVI_{r}(F, x) \geq HVI_{r}(F, s) > 0,$$

we obtain there exists no $f \in F$ with $f \preceq p$ by Corollary A4. i.e., for all $f \in F$ there exists $i_f$ with $p_{i_f} < f_{i_f}$. Assume there exists $x \in S$ with $x \prec p$ and $x_i < p_i$. Then, there exists no $f \in F$ with $f \preceq x$ since $x \preceq p$. By applying Corollary A5, we deduce

$$HVI_{r}(F \cup \{x\}) > HVI_{r}(F \cup \{p\}) \geq HVI_{r}(F)$$

which contradicts $HVI_{r}(F, p) = \max_{x \in S} HVI_{r}(F, x)$. □

Appendix B. Probability Measure for Multivariate Normal Distribution

**Theorem A6.** Let $I$ be a set and for every $J \subset I$ finite an inner regular probability measure $P_J$ on $\mathbb{R}^J$ be given. Given two finite $J \subset J' \subset I$, denote by $pr_{J \subset J'}: \mathbb{R}^{J'} \rightarrow \mathbb{R}^J$ the canonical projection. Assume that for all $J \subset J' \subset I$ finite

$$P_J \circ pr_{J \subset J'}^{-1} = P_J$$

holds. Then, there exists a unique measure $P_I$ on $\mathbb{R}^I$ satisfying

$$P_I \circ pr_{J \subset I}^{-1} = P_J$$

for all $J \subset I$ finite.

Before giving the proof, recall:

**Theorem A7 (Hahn-Kolmogorov).** Let $S$ be a set and $R \subset \mathcal{P}(S)$ be ring, i.e., $\emptyset \in R$ and $R$ is stable under finite unions and binary complements. Let

$$P_0: R \rightarrow [0, \infty)$$

be a pre-measure, i.e.,

$$P(\emptyset) = 0$$

and

$$P_0(\bigcup_{n \in \mathbb{N}} A_i) = \sum_{n \in \mathbb{N}} P_0(A_i)$$

for pairwise disjoint

$$A_1, A_2, \ldots \in R$$

with

$$\bigcup_{n \in \mathbb{N}} A_i \in R.$$

Then, $P_0$ extends to a measure $P$ on the sigma algebra $\sigma(R)$ generated by $R$. Furthermore, if $P_0$ is $\sigma$-finite, then, the extension is unique.

**Proof of Theorem A6.** The sigma algebra of the product $\mathbb{R}^I$ is generated by

$$R = \{ A \times \mathbb{R}^{I-J} | J \subset I \text{ finite}, A \in \mathcal{B}(\mathbb{R}^J) \}.$$ 

The reader convinces himself that $R$ is a ring. Define a function

$$P_0: R \rightarrow [0, \infty)$$
by $P_0(A \times \mathbb{R}^{l-j}) = P_j(A)$. Observe that given $A \times \mathbb{R}^{l-j} = A' \times \mathbb{R}^{l-j'}$, then, without loss of generality $j \subset j'$ and $A' = A \times \mathbb{R}^{j-j}$. Thus,

$$
P_0(A' \times \mathbb{R}^{j-j'}) = P_{j'}(A')
$$

by definition

$$
= P_{j'}(A \times \mathbb{R}^{j-j'})
= P_{j'} \circ pr_{j-j'}^{-1}(A)
= P_j(A)

P_{j'} \circ pr_{j-j'}^{-1} = P_j
$$

and $P_0$ is well defined. Then, the reader convinces himself that

$$P_0(\emptyset) = P_j(\emptyset) = 0$$

and $P_0$ is finite additive. We prove that $P_0$ is $\sigma$-additive, i.e., given $A_1, ..., \in R$ pairwise disjoint with $\bigcup_{n \in \mathbb{N}} A_n \in R$, then,

$$P_0(\bigcup_{n \in \mathbb{N}} A_n) = \sum_{n \in \mathbb{N}} P_0(A_n).$$

Then, the Hahn-Kolmogorov theorem above proves the claim. Notice that every probability measure is $\sigma$-finite and, thus, $P_0$ is.

It is well known that it suffices to prove

$$\lim_{n \in \mathbb{N}} P_0(A_n) = 0$$

for all

$$A_1 \supset A_2 \supset \cdots \in R$$

with $\cap_{n \in \mathbb{N}} A_n = \emptyset$ since $P_0$ is finite additive. We prove that if there exists an $\epsilon > 0$ such that

$$P_0(A_n) > 2\epsilon$$

for all $n \in \mathbb{N}$, then, $\cap_{n \in \mathbb{N}} A_n \neq \emptyset$. Write $A_i = B_i \times \mathbb{R}^{l-l_i}$ such that $I_i \subset I_{i+1}$ for every $i$. Then,

$$P_0(A_i) = P_{j_i}(B_i) = \sup_{K \subset B_i \text{ compact}} P_{j_i}(K)$$

since $P_{j_i}$ is inner regular. For all $i$ choose some $K_i \subset B_i$ compact such that

$$P_{j_i}(B_i) - P_{j_i}(K_i) < 2^{-i}\epsilon.$$  \hspace{1cm} (A2)

Write $C_i = K_i \times \mathbb{R}^{l-l_i}$ for every $i$. We first prove

$$P_0(\cap_{i=1}^n C_i) > \epsilon$$

for all $n$. Notice

$$P_0(\cap_{i=1}^n C_i) = P_0(\cap_{i=1}^n A_i) - P_0(\cap_{i=1}^n A_i - \cap_{i=1}^n C_i)$$

since $P_0$ is finite additive and

$$\cap_{i=1}^n C_i \cup (\cap_{i=1}^n A_i - \cap_{i=1}^n C_i) = \cap_{i=1}^n A_i.$$  \hspace{1cm} (A2)

Furthermore, $\cap_{i=1}^n A_i = A_n$ since $A_{i+1} \subset A_i$ for all $i$. Thus,

$$P_0(\cap_{i=1}^n A_i) = P_0(A_n) > 2\epsilon.$$
by (A2) and it suffices to prove
\[ P_0(\bigcup_{i=1}^n A_n - C_i) = P_0(\bigcap_{i=1}^n A_i - \bigcap_{i=1}^n C_i) < \epsilon \]
for all \( n \). It holds
\[
P_0(\bigcup_{i=1}^n A_n - C_i) \leq P_0(\bigcap_{i=1}^n A_i - C_i) \leq \sum_{i=1}^n (P_0(A_i) - P_0(C_i))
\]
by definition
\[
= \sum_{i=1}^n (P_j(B_i) - P_j(K_i)) \leq \sum_{i=1}^n 2^{-i} \epsilon \quad (A2)
\]
In particular, \( \bigcap_{i=1}^n C_i \neq \emptyset \) and, hence, \( D_n := \bigcap_{i=1}^n K_i \neq \emptyset \) for all \( n \). We consider the descending sequence
\[ D_1 \supset D_2 \supset \cdots \in \mathbb{R}^t \]
Since all \( D_i \) are compact, \( \mathbb{R}^t \) is Hausdorff (hence, \( D_1 \) is) and all \( D_i \) are non-empty, the below claim ensures that
\[ \emptyset \neq \bigcap_{i \in \mathbb{N}} K_i \subset \bigcap_{i \in \mathbb{N}} A_i \]
and, hence, the claim. \( \square \)

**Lemma A8.** Let
\[ E \supset K_0 \supset K_1 \supset \cdots \]
be a descending sequence of compact topological spaces with \( E \) being a Hausdorff space.
Then \( K = \bigcap_{i \in \mathbb{N}} K_i = \emptyset \) implies that there exists some \( n \in \mathbb{N} \) with \( K_n = \emptyset \).

**Proof.** If \( K = \emptyset \), then, its complement
\[ E = K^c = E - K = \bigcup_{n \in \mathbb{N}} (E - K_i) \]
is \( E \). Recall every compact subset of a Hausdorff space to be closed. Hence, \( E - K_i \) is open for every \( i \). Since \( E \) is compact, there exist \( i_1, \ldots, i_k \) such that \( E = \bigcup_{j=1}^{i_1} (E - K_j) \). Hence, \( \bigcap_{j=1}^{i_1} K_j = \emptyset \) is empty. Thus,
\[ K_n \subset \bigcap_{j=1}^{i_1} K_j = \emptyset \]
for some \( n \) greater all \( i_j \). \( \square \)

**Proof of Theorem 3.** In view of Theorem A6, it suffices to prove
\[
\mathcal{N}(m_{F'}, C_{F'}) = \mathcal{N}(m_{F}, C_{F}) \circ pr_{F' \subset F}^{-1}
\]
for any finite subsets \( F' \subset F \subset X \). Observe that \( pr_{F' \subset F} : \mathbb{R}^F \to \mathbb{R}^{F'} \) is given by left multiplication with the matrix \( A = (\delta_{ij})_{i \in F', j \in F} \) and that \( A \) has full rank (since the projection is an epimorphism). Furthermore, by construction we obtain equalities
\[ C_{F'} = AC_F A^T \]
and
\[ m_{F'} = Am.F. \]

Then, the claim is precisely “(9.5) Satz” in [25] for \( A = A \) and \( a = 0 \). \( \square \)

**Proof of Corollary 6.** Denote by
\[ C_{j,F} = (C(j,f))_{j \in I, f \in F}. \]

Given some \( J \subset X \) finite, a straightforward computation proves
\[
m'_{j} := (m'(j))_{j \in I} = m_{j} + C_{j,F}C^{-1}_{F}(y - m_{F})
\]
\[
C'_{j} := (C'(j_{1},j_{2}))_{j_{1},j_{2} \in I} = C_{j} + C_{j,F}C^{-1}_{F}C_{F,j}.
\]

Then, \( m' \) and \( C' \) are functions such that for every \( I \subset X \) finite the induced matrix \( C'_{j} = (C'(j_{1},j_{2}))_{j_{1},j_{2} \in I} \) is positive definite and the density function of the induced normal distribution
\[ N(m'_{j},C'_{j}) \]

is the conditional density function induced by \( ((r_{j})_{j \in I}, (r_{f})_{f \in F}) \) given \( (r_{f})_{f \in F} = y \) by Theorem 2. Applying Theorem 3 with mean function \( m' \) and covariance function \( C' \) finishes the proof. \( \square \)

**Appendix C. Integrability of Hypervolume Improvement**

**Lemma A9.** The Hypervolume improvement function
\[ HVI_{r,f}: S \rightarrow \mathbb{R}, y \mapsto HVI_{r}(F, y) \]

is continuous, hence, integrable for \( F \) finite and \( S \subset \mathbb{R}^{d} \) such that \( s_{n} < r \) for all \( s \in S, f \in F \).

**Proof.** It suffices to prove that
\[ h: \mathbb{R}^{d} \rightarrow \mathbb{R}, y \mapsto HV_{r}(F \cup \{ y \}) \]

is continuous. First, recall that given sets \( F_{1}, ..., F_{n} \), the Lebesgue measure of their union is given by
\[
\lambda(\bigcup_{i=1}^{n} F_{i}) = \sum_{i=1}^{n} \lambda(F_{i}) - \sum_{1 \leq i < j \leq n} \lambda(F_{i} \cap F_{j}) + \\
\quad + \sum_{1 \leq i < j < k \leq n} \lambda(F_{i} \cap F_{j} \cap F_{k}) + \cdots \\
\quad \quad \quad + (-1)^{n-1} \lambda(\bigcap_{i=1}^{n} F_{i})
\]
since \( \lambda \) is additive. Given \( s_{1}, ..., s_{n} \in \mathbb{R}^{d} \), we define
\[
i(s_{1}, ..., s_{n}) = (\max(s_{1}, ..., s_{n}))_{i=1,...,n}.
\]

Clearly, \( I(a,r) \cap I(b,r) = I(i(a,b),r) \) for all \( a, b < r \). Using \( i(s_{1}, ..., s_{n}) = i(i(s_{1}, ..., s_{n-1}), s_{n}) \), we obtain
\[
\cap_{i=1}^{n} I(s_{i}, r) = I(i(s_{1}, ..., s_{n}), r)
\]

for \( s_{i} < r \) by induction. Writing
\[ h(y) = \lambda(\bigcup_{f \in F} I(f,r) \cup I(y,r)) \]
as sum as above and using (A4), it suffices to prove
\[ g : S \rightarrow \mathbb{R}, y \mapsto \lambda(I(i(s_1, ..., s_n, y), r)) \]
to be continuous for all \( s_1, ..., s_n \in S \). Therefore, it suffices to prove
(i)
\[ i(s_1, .., s_n, -) : \mathbb{R}^l \rightarrow \mathbb{R}^l, y \mapsto i(s_1, .., s_n, y) \]
and
(ii)
\[ \lambda(I(-, r)) : S \rightarrow \mathbb{R}, y \mapsto \lambda(I(y, r)) \]
to be continuous.
To (i):
We observe
\[ i(s, y) = \frac{1}{2}(s + y + |s - x|) \]
(where everything is taken componentwise) and, thus, \( i(s, -) \) is continuous for all \( s \). Using
\[ i(s_1, ..., s_n, y) = i(i(s_1, ..., s_n), y), \]
we obtain the claim.
To (ii):
We calculate
\[ \lambda(I(y, r)) = \prod_{i=1}^{l}(r_i - y_i) \]
which is continuous in \( y \).

References
1. Elkington, R. Leadership Decision-Making Leveraging Big Data in Vuca Contexts. J. Leadersh. Stud. 2018, 12, 66–70. [CrossRef]
2. Afshari, H.; Hare, W.; Tesfamariam, S. Constrained multi-objective optimization algorithms: Review and comparison with application in reinforced concrete structures. Appl. Soft Comput. 2019, 83, 105631. [CrossRef]
3. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Proc. IEEE 2016, 104, 148–175. [CrossRef]
4. Lyu, W.; Xue, P.; Yang, F.; Yan, C.; Hong, Z.; Zeng, X.; Zhou, D. An Efficient Bayesian Optimization Approach for Automated Optimization of Analog Circuits. IEEE Trans. Circuits Syst. I Regul. Pap. 2018, 65, 1954–1967. [CrossRef]
5. Zhang, S.; Yang, F.; Yan, C.; Zhou, D.; Zeng, X. An Efficient Batch-Constrained Bayesian Optimization Approach for Analog Circuit Synthesis via Multiobjective Acquisition Ensemble. IEEE Trans.- Comput.-Aided Des. Integr. Circuits Syst. 2022, 41, 1–14. [CrossRef]
6. Guo, J.; Crupi, G.; Cai, J. A Novel Design Methodology for a Multiotcave GaN-HEMT Power Amplifier Using Clustering Guided Bayesian Optimization. IEEE Access 2022, 10, 52771–52781. [CrossRef]
7. Sawant, M.M.; Bhurchandi, K. Hierarchical Facial Age Estimation Using Gaussian Process Regression. IEEE Access 2019, 7, 9142–9152. [CrossRef]
8. Huang, H.; Song, Y.; Peng, X.; Ding, S.X.; Zhong, W.; Du, W. A Sparse Nonstationary Trigonometric Gaussian Process Regression and Its Application on Nitrogen Oxide Prediction of the Diesel Engine. IEEE Trans. Ind. Inform. 2021, 17, 8367–8377. [CrossRef]
9. Koriyama, T.; Nose, T.; Kobayashi, T. Statistical Parametric Speech Synthesis Based on Gaussian Process Regression. IEEE J. Sel. Top. Signal Process. 2014, 8, 173–183. [CrossRef]
10. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. J. Math. Psychol. 2018, 85, 1–16. [CrossRef]
11. Lewis-Beck, C.; Lewis-Beck, M. Applied Regression: An Introduction; Sage Publications: Thousand Oaks, CA, USA, 2015; Volume 22.
12. Verleysen, M.; François, D. The curse of dimensionality in data mining and time series prediction. In Proceedings of the International Work-Conference on Artificial Neural Networks, Barcelona, Spain, 8–10 June 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 758–770.
13. Frazier, P.I. Bayesian optimization. In Recent Advances in Optimization and Modeling of Contemporary Problems; Informs: Phoenix, AZ, USA, 2018; pp. 255–278. [CrossRef] [PubMed]
14. Emmerich, M.; Deutz, A.H. A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. Nat. Comput. 2018, 17, 585–609. [CrossRef] [PubMed]
15. Anderson, T.W. An Introduction to Multivariate Statistical Analysis, 3rd ed.; Wiley: New York, NY, USA, 2003.
16. Duvenaud, D. Automatic Model Construction with Gaussian Processes. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2014.

17. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. Reference point specification in hypervolume calculation for fair comparison and efficient search. In Proceedings of the Genetic and Evolutionary Computation Conference, Berlin, Germany, 15–19 July 2017; pp. 585–592. [CrossRef]

18. Emmerich, M.; Deutz, A.; Klinkenberg, J. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation (CEC), Ritz Carlton, New Orleans, LA, USA, 5–8 June 2011; pp. 2147–2154. [CrossRef]

19. Yang, K.; Emmerich, M.; Deutz, A.; Bäck, T. Efficient computation of expected hypervolume improvement using box decomposition algorithms. J. Glob. Optim. 2019, 75, 3–34. [CrossRef]

20. McKay, M.D.; Beckman, R.J.; Conover, W.J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics 1979, 21, 239. [CrossRef]

21. Emmerich, M. KMAC V1.0 - The efficient O(n log n) implementation of 2D and 3D Expected Hypervolume Improvement (EHVI). Available online: https://liacs.leidenuniv.nl/~csmoda/index.php?page=code (accessed on 22 June 2022).

22. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 2002, 6, 182–197. [CrossRef]

23. Siemens, Data Sheet PSS®DE. Available online: https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/grid-edge-software/pssde.html (accessed on 24 August 2022).

24. Proedrou, E. A comprehensive review of residential electricity load profile models. IEEE Access 2021, 9, 12114–12133. [CrossRef]

25. Georgii, H.O. Stochastik: Einführung in die Wahrscheinlichkeitsrechnung und Statistik, 5. Auflage; Walter de Gruyter: Berlin, Germany, 2007.