Cheap-expensive multi-objective Bayesian optimization for permanent magnet synchronous motor design

Peer-reviewed author version

Loka, Nasrulloh; Ibrahim, Mohamed; Couckuyt, Ivo; VAN NIEUWENHUYSE, Inneke & Dhaene, Tom (2023) Cheap-expensive multi-objective Bayesian optimization for permanent magnet synchronous motor design. In: ENGINEERING WITH COMPUTERS,.

DOI: 10.1007/s00366-023-01900-0
Handle: http://hdl.handle.net/1942/41753

Cheap-Expensive Multi-Objective Bayesian Optimization for Permanent Magnet Synchronous Motor Design

Nasrulloh Loka[1,6*], Mohamed Ibrahim[2,4,5], Ivo Couckuyt[1], Inneke Van Nieuwenhuyse[3] and Tom Dhaene[1]

[1*]Department of Information Technology (INTEC), IDLab, Ghent University-imec, iGent, Technologiepark-Zwijnaarde 126, Ghent, B-9052, Belgium.
[2] Department of Electromechanical, Systems and Metal Engineering, Ghent University, Technologiepark-Zwijnaarde 131, Ghent, B-9052, Belgium.
[3]FlandersMake@UHasselt and Data Science Institute, Hasselt University, Martelarenlaan 42, Hasselt, 3500, Belgium.
[4] FlandersMake@UGent—Corelab EEDT-MP, Leuven, 3001, Belgium.
[5] Electrical Engineering Department, Kafrelshiekh University , Kafrelshiekh, 33511, Egypt.
[6]Research Group Logistics, Hasselt University, Agoralaan Gebouw D, Diepenbeek, Limburg, 3590, Belgium.

*Corresponding author(s). E-mail(s): nasrulloh.loka@ugent.be;
Contributing authors: mohamed.ibrahim@ugent.be; ivo.couckuyt@ugent.be; inneke.vannieuwenhuyse@uhasselt.be; tom.dhaene@ugent.be;

Abstract

Bayesian Optimization (BO) is a popular optimization technique for expensive-to-evaluate black-box functions. We propose a cheap-expensive multi-objective BO strategy for optimizing a Permanent Magnet Synchronous Motor (PMSM). The design of an electric motor is a complex, time-consuming process that contains various heterogeneous objectives

and constraints; in particular, we have a mix of cheap and expensive objective and constraint functions. The expensive objectives and constraints are usually quantified by a time-consuming finite element method, while the cheap ones are available as closed-form equations. We propose a BO policy that can accommodate cheap-expensive objectives and constraints, using a hypervolume-based acquisition function that combines expensive function approximation from a surrogate with direct cheap evaluations. The proposed method is benchmarked on multiple test functions with promising results, reaching competitive solutions much faster than traditional BO methods. To address the aforementioned design challenges for PMSM, we apply our proposed method, which aims to maximize motor efficiency while minimizing torque ripple and active mass, and considers six other performance indicators as constraints.

Keywords: Bayesian Optimization, Multi-objectives Optimization, Constrained Optimization, Permanent Magnet Synchronous Motor

1 Introduction

Bayesian Optimization (BO) [1–4] is a popular surrogate-based data-efficient technique for optimizing complex and time-consuming optimization problems [5, 6]. It is particularly useful when data is limited or expensive to acquire. This paper presents a case study in which both cheap and expensive objective and constraint functions are considered in the design of electric motors.

In the case of electric motor design, many geometric and electromagnetic parameters can affect the motor's performance. BO can help to efficiently identify the optimal combination of these parameters to achieve the desired performance indicators (such as high efficiency and high torque density, as required for use in electric vehicles).

Electric motor design optimization is highly relevant in practice, as electric motors consume about 40% of the generated energy worldwide [7]. Usually, the optimization is done using a Genetic Algorithm (GA) evaluated on Finite Element Methods (FEM), requiring large numbers of evaluations [8]. Such FEM evaluations can take hours to days, depending on the geometries of the motor under study; consequently, this design optimization problem could benefit substantially from an efficient optimization method [7, 9, 10]. Our approach uses a surrogate model to approximate the expensive FEM evaluations [8, 11, 12]. Objectives and constraints that can be calculated cheaply (i.e., without the need for FEM, such as the total mass of the material) do not require a surrogate model, though; they can be quantified or approximated using deterministic closed-form formulas. Our approach distinguishes between the cheap and expensive functions in the optimization procedure; we show that this yields substantial improvements in data efficiency compared to traditional BO methods.

BO has two core components: a surrogate model and an acquisition function. The surrogate model is used to approximate the expensive output functions (either objectives or constraints) cheaply. The choice for the surrogate model is commonly a Gaussian Process (GP) [13, 14] or any other statistical model with uncertainty quantification capability such as Polynomial Chaos Expansion [15, 16], Neural Networks [17, 18], or Tree Parzen Estimators [19]. Based on the model prediction and the uncertainty quantification, an acquisition function is defined to sequentially search for the optimum design by balancing exploration and exploitation. A lot of BO research is available, accounting for different complexities in the optimization setting, such as batch optimization [20, 21], multi-fidelity [22], constrained optimization [23, 24], and multiple objectives [25, 26]. A review paper discussing problem settings in BO is presented in [27].

In Multi-Objective Bayesian Global Optimization (MOBGO), cheap and expensive objectives are commonly treated in the same manner, i.e., modeled using surrogate models. Some attempts to exploit cheap-expensive properties are presented in recent literature: Allmendinger et al. [28] extend the genetic algorithm approach to deal with cheap objectives by using a fast-first and interleaving method. Wang et al. [29–31] study the relationship between cheap-expensive objectives and search bias in evolutionary algorithm settings. Loka et al. [32] propose a hypervolume-based BO approach considering a mix of cheap and expensive objectives, but only for an unconstrained bi-objective setting.

In this study, a two-stage constrained MOBGO algorithm is presented to optimize a Permanent Magnet Synchronous Motor (PMSM) design. The algorithm explicitly distinguishes between the cheap and expensive output functions. The *first stage* is a constrained active learning (AL) step used to improve the accuracy of the expensive constraint predictions. This is especially useful when these constraint functions are highly irregular (showing many local optima) and/or when the feasible region of the solution space is very small (implying that the initial design may contain only a few or even no feasible designs). The *second stage* is the optimization stage, which uses the proposed hypervolume-based cheap-expensive constrained acquisition function (CEHVI-C). This function extends the work of Yang et al. [33] by incorporating the cheap *objectives* directly in the hypervolume calculation. The *cheap constraints* are accounted for in the optimizer of the BO acquisition function. We show that the resulting algorithm can attain competitive solutions faster than the traditional BO method.

The key contributions of this paper are the following:

- The proposed approach builds on a flexible way to quantify hypervolume, exploiting the distinction between cheap and expensive objectives. This improves the computational effort for calculating this metric. Moreover, contrary to the work in [33], it is applicable with any arbitrary box decomposition approach. Additionally, the algorithm handles expensive and cheap constraints in a clearly distinct way (accounting for the former in the probability of feasibility and for the latter in the optimization of the acquisition

function). As shown, this results in an algorithm that is data-efficient and yields high-quality solutions.
- Using the proposed approach, we show that the PMSM design problem can be solved in a data-efficient manner, which outperforms the common approaches used to solve this problem in the literature. This is in itself an insightful result, as the FEM calculations are very expensive.

The rest of this paper is organized as follows: Section 2 describes the PMSM under study. Section 3 explains the basics of multi-objective optimization in general, along with the corresponding notation and terminology. Section 4 presents the proposed algorithm for constrained multi-objective problems with cheap and expensive outcomes. In section 5, the experimental setup and results are discussed. Finally, section 6 presents the conclusions of this paper.

2 Permanent Magnet Synchronous Motor Optimization

A permanent magnet synchronous motor (PMSM) has several advantages compared to other types of electric motors, such as higher power density and higher efficiency. Consequently, this type of motor is preferred in settings where power density and efficiency are critical, such as in automotive applications. The downside is that this motor uses rare-earth magnets, which are not only very expensive but also unfriendly for the environment because of the recycling problems [34, 35] and the impact of the mining activities [36].
Figure 1 shows a schematic drawing of the motor with the relevant geometrical design parameters, which are further detailed in Table 1. The magnets (referred to as rotor poles, pm) are located on the rotor in red and black (to reflect different polarities).

Table 1: Geometrical design parameters of the PMSM design optimization problem

Design Variable	Type	Unit	Value
(1) Stator outer radius (SOR)	Constant	mm	96
(2) Rotor yoke thickness (RYt)	Variable	mm	5.0-20.0
(3) Width of pm (Wpm)	Variable		0.7-0.9
(4) Thickness of pm (Thpm)	Variable	mm	3.0-5.0
(5) Rotor outer radius (ROR)	Variable	mm	35.0-65.0
(6) Slot height (sh)	Variable	mm	10.0-25.0
(7) Slot teeth ratio (STR)	Variable		0.4-0.7
(8) Axial length (Lfe)	Variable	mm	40.0-60.0
(9) Number of pm (p)	Constant		10

The PMSM design problem has three objectives (see Table 2) and six constraints (see Table 3).

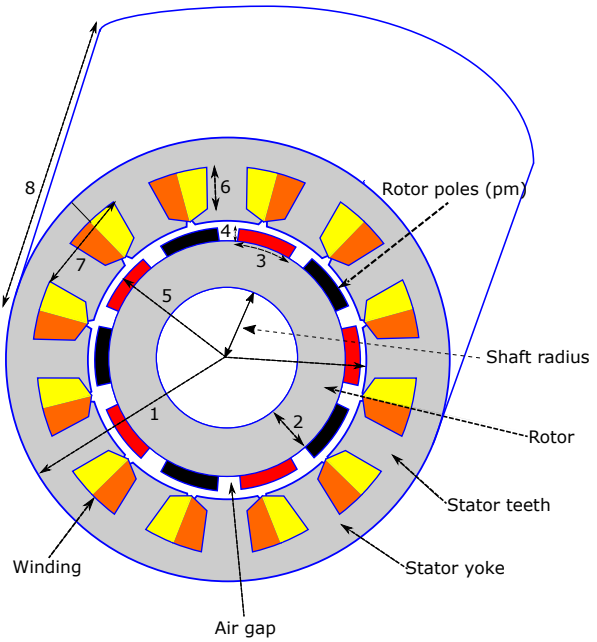


Fig. 1: Motor geometry with geometrical annotations

The optimization of these parameters is nontrivial, though, as they tend to have conflicting impacts on the objectives: increasing the axial length of the motor (parameter (8) in Fig. 1), for instance, increases the average torque but simultaneously increases the total mass (and hence, cost). Consequently, this design optimization problem is a constrained multi-objective optimization problem. Some of the performance indicators in Tables 2 and 3 are cheap, meaning that they can be calculated by means of closed-form formulas. The expensive indicators are evaluated using Finite Element Methods (FEM) [37]. More details on the performance indicators can be found in Appendix 6.

Table 2: Objectives for the Optimization of a PMSM

Objective Name	Type	Unit	Optimization Type
Efficiency	Expensive	%	Maximization
Torque ripple	Expensive	N.m	Minimization
Total mass	Cheap	kg	Minimization

The PMSM problem is similar to the one proposed in [38, 39]. Some work has been done to perform multi-objective optimization on a similar motor design, but it relies on many FEM evaluations [39–41] and handcrafted optimization steps [39, 42].

Table 3: Constraints for the Optimization of a PMSM

Constraint Name	Type	Constraint
Magnitude of flux density in stator yoke (BmagSY)	Expensive	$g_1 \leq 1.5$ T
Magnitude of flux density in stator teeth (BmagST)	Expensive	$g_2 \leq 1.5$ T
Thermal loading	Expensive	$g_3 \leq 8$ kW/m^2
Average torque (Tavg)	Expensive	$60Nm \leq g_4 \leq 65Nm$
Shaft radius	Cheap	$g_5 \geq 15mm$
Stator yoke thickness	Cheap	$g_6 \geq 5mm$

This study focuses on developing a data-driven approach that also minimizes the number of expensive evaluations, and so that it can be applied to different problems and settings.

3 Constrained MOO: Problem Formulation

The goal of a constrained Multi-Objective Optimization (MOO) method is to optimize a set of objective functions $\boldsymbol{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})] \in \mathbb{R}^M$, while satisfying a set of constraints $\boldsymbol{g}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_V(\mathbf{x})] \leq 0 \in \mathbb{R}^V$:

$$\begin{aligned} &\min_{\mathbf{x}} \left(f_1(\mathbf{x}), \ldots, f_M(\mathbf{x})\right) \\ &\text{s.t. } g_v(\mathbf{x}) \leq 0, \quad v = 1, \ldots, V \end{aligned} \tag{1}$$

where $M \geq 2$ is the number of objectives, $V \geq 1$ is the number of constraints, and $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$. The set $\mathcal{X}$ is d-dimensional and bounded. Without loss of generality, this paper assumes that the objectives need to be minimized (except when explicitly stated otherwise). In MOO, there typically is no *single* optimal solution $\mathbf{x}^*$ that minimizes all objectives simultaneously while satisfying all constraints; instead, there is a *set of optimal solutions*, referred to as the Pareto set. Mathematically, the Pareto set for an *unconstrained* optimization problem is defined as:

$$\boldsymbol{P} = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{x}' \in \mathcal{X} : \mathbf{x}' \prec \mathbf{x}\} \tag{2}$$

where the notation $\mathbf{x}_b \prec \mathbf{x}_a$ means that $\mathbf{x}_b$ *dominates* $\mathbf{x}_a$. In a minimization problem with M objectives, $\mathbf{x}_b \prec \mathbf{x}_a$ if and only if $f_m(\mathbf{x}_b) \leq f_m(\mathbf{x}_a), \forall m \in \{1, .., M\}$ and $\exists m \in \{1, .., M\}$ such that $f_m(\mathbf{x}_b) < f_m(\mathbf{x}_a)$. Informally, we can say that $\mathbf{x}_b$ dominates $\mathbf{x}_a$ if and only if it is better in at least one objective, while not being worse in any of the other objectives. As evident from Eq. (2), $\boldsymbol{P}$ is defined in the input space; the image of the Pareto set in the objective space is referred to as the Pareto front: $\mathcal{P} = \left\{\boldsymbol{f}(\mathbf{x}) \in \mathbb{R}^M \mid \nexists \mathbf{x}' \in \mathcal{X} : \mathbf{x}' \prec \mathbf{x}\right\}$. In *constrained* problems, only feasible points $\mathbf{x}$ can be part of the Pareto set. We thus define Pareto feasible set as:

$$\boldsymbol{P}_{feas} = \{\mathbf{x} \in \mathcal{X} \mid \nexists \mathbf{x}' \in \mathcal{X} : \mathbf{x}' \prec \mathbf{x},\ \boldsymbol{g}(\mathbf{x}) \leq 0,\ \boldsymbol{g}(\mathbf{x}') \leq 0\} \tag{3}$$

For ease of notation, we denote $\boldsymbol{P} := \boldsymbol{P}_{feas}$ and $\mathcal{P} := \mathcal{P}_{feas}$ for every constrained problem in this paper.

In this work, Bayesian Optimization (BO) is used to find the Pareto set in a data-efficient manner (i.e., using the smallest possible number of function evaluations). Bayesian optimization has two main components: (1) the surrogate model, which approximates the expensive output functions, and (2) the acquisition function, which guides the BO procedure by sequentially selecting additional input points to evaluate. BO automatically balances exploration and exploitation.[13, 14, 43]. The Gaussian Process (GP) is the most popular type of surrogate model used in BO; the technical details of the model can be found in Appendix B.1. The proposed acquisition function is a key component and is discussed in the following section.

4 MOBGO algorithm for Cheap-Expensive Objectives and Constraints

Previous research on MOBGO algorithms commonly uses an acquisition function based on the hypervolume metric to search for the Pareto optimal points [25, 26, 33, 44]. Very recently, this type of acquisition function has been applied in mixed-variable settings [45], parallel evaluation settings [46], and for high-dimensional problems [47]. Yet, none of these previous works exploit potential differences in the latencies (i.e., the evaluation times) of the different objective functions. In real-life problems, it often occurs that the output functions (objectives and/or constraints) are a mix of cheap and expensive functions.

To the best of our knowledge, the only papers available so far on this topic are [48, 49]

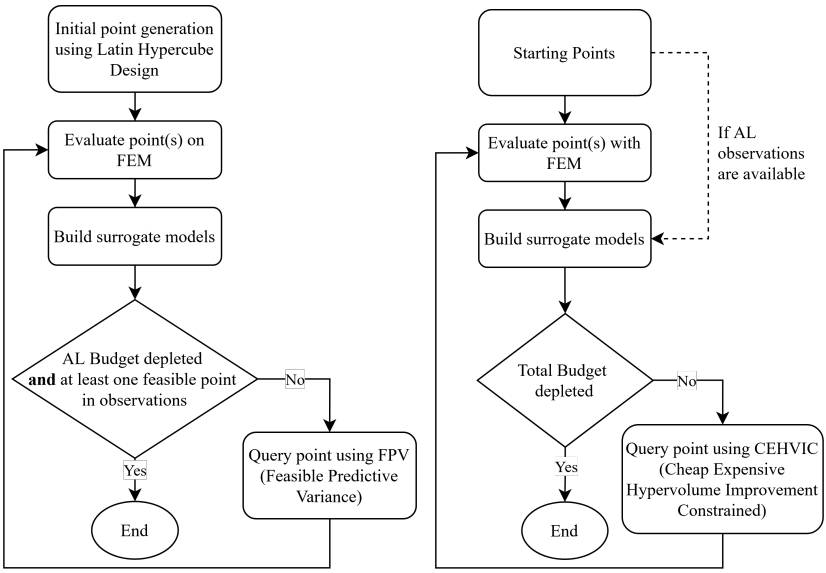
(which focus on exploiting latency differences in evolutionary algorithms), and [32] (which presents a BO algorithm limited to bi-objective unconstrained MOO problems).

Recently, Buckingham et al. [50] proposed a scalarization-based multi-objective BO approach for a similar problem. However, their method assumes that the cheap objective does not have a closed-form formula.

We propose a two-stage optimization approach (as in [51]), which is depicted in Figure 2. The first stage is optional and is referred to as the Active Learning (AL) stage. It aims to improve the accuracy of the GPs for hard-to-model constraints (if any), using the Feasible Predictive Variance acquisition function discussed in Section 4.1. In the AL phase, the initial surrogates for these constraints are estimated based on a set of initial design points, which are evaluated using the expensive FEM models. The most common choice in the BO literature is a maximin Latin Hypercube design, [52] to ensure that the resulting set is space-filling. As the aim of the AL phase is to improve the accuracy of these constraint models, additional points are queried using the Feasible Predictive Variance acquisition function, which is discussed in detail in Section 4.1. The AL stage ends when the specified AL budget is depleted *and* there is at least one feasible point present in the observations. In some cases, the feasible area of the problem is very small, which may force the analyst to keep querying points until *both* conditions are fulfilled.

The second stage is the Bayesian optimization stage. If it was preceded by the AL stage, the resulting observations are used as starting points, on which the surrogate models are again estimated. If the AL phase was skipped, the starting points are generated through a space-filling design (usually a maximin Latin Hypercube design, for the reasons stated above), and they are first evaluated with the expensive FEM models in order to estimate the initial surrogate models. The additional points to evaluate are then selected using the proposed cheap-expensive hypervolume-based expected improvement acquisition function, which is discussed in Section 4.2. The algorithm ends when the total budget has been depleted, and the points that have been evaluated as feasible and Pareto-optimal are put forward as the points on the front.

As explained below, the acquisition functions of both stages account for the estimated Probability of Feasibility (PF) of the point with respect to the *expensive* constraints, to avoid spending the budget for evaluating points that are likely infeasible. The feasibility with respect to the *cheap* constraints is handled inside the optimization procedure that maximizes the acquisition functions, as discussed in Section 4.2.2.



(a) Active learning (b) Bayesian Optimization

Fig. 2: Two-stage optimization scheme. The first stage (AL) is optional. We use it in our experiments for the PMSM case. The data set resulting from the active learning steps in (a) are then used as the starting points in (b). If the AL phase is skipped, the starting points in (b) are generated by a Latin Hypercube design.

4.1 Stage 1: Active learning (AL)

This stage aims to improve the accuracy of the GP models of the expensive constraints by focusing solely on exploring the region where the model exhibits high uncertainty. This is especially useful when the expensive constraint functions are non-smooth (i.e., highly irregular, showing many local optima, which makes the function hard to model) and/or when the feasible region of the solution space is very small (implying that the initial design may contain only a few or even no feasible designs). In such cases, the information gained during the AL stage results in significant efficiency gains in the optimization stage.

The acquisition function used is the Feasible Predictive Variance (FPV), which is defined as:

$$FPV = \prod_{v=1}^{V} \sigma_v^2 \cdot PF_v(\mathbf{x}_*) \tag{4}$$

where σ_v^2 is the predictive variance of the *hard-to-model* constraint v at x_*, and PF_v refers to the Probability of Feasibility (see e.g. [23]) of x_* for constraint v:

$$PF_v(\mathbf{x}_*) := \Pr[\tilde{g_v}(\mathbf{x}_*) \leq 0] \tag{5}$$

$$= \int_{-\infty}^{0} p\left(\tilde{g_v}(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{T}_N\right) d\tilde{g_v}(\mathbf{x}_*) \tag{6}$$

$\tilde{g_v}(\mathbf{x}_*)$ refers to the Gaussian process outcome for constraint v at $\mathbf{x}_*$, and $\mathcal{T}_N$ denotes the data set already available for constraint v.

In the PMSM case study, we use this stage specifically for the average torque constraint (see Table 3) as this constraint is hard to model, and moreover it restricts the number of feasible solutions more severely than the other constraints.

4.2 Stage 2: Bayesian Optimization

After finding enough feasible solutions, new points are selected by maximizing the Cheap-Expensive Expected Hypervolume Improvement with Constraints (CEHVI-C) acquisition function:

$$\text{CEHVI-C} = \text{CEHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) \cdot \prod_{v=1}^{V} PF_v(\mathbf{x}_*) \tag{7}$$

CEHVI-C multiplies the proposed CEHVI acquisition function (see Section 4.2.1) with the probability of feasibility of *all* expensive constraints g_v ($v = 1 \cdots V$). Assuming that all these constraints are independent, this reduces to a multiplication of the individual $PF_v(\mathbf{x}_*)$. Note that equation 7 also implicitly assumes conditional independence between the objective and constraint functions [23]. Both assumptions are standard in constrained Bayesian optimization.

The *cheap constraints* are not directly incorporated in equation 7, since this would likely introduce severe non-smooth behavior in the response surface of the acquisition function. Instead, the cheap constraints are accounted for in the optimization procedure implemented to maximize the acquisition function, which is further detailed in Section 4.2.2.

4.2.1 Cheap-Expensive Expected Hypervolume Improvement

We extend the EHVI formulation presented in [33] such that (1) it can efficiently handle a mix of cheap and expensive objectives, and (2) it can be used independent of the hyperbox decomposition method chosen to implement the calculations.

In hypervolume-based MOBGO, the notion of improvement by the Lebesgue measure is used. Let us first define the hypervolume indicator (HVI) $\mathcal{H}$ [53, 54]. Given a Pareto front $\mathcal{P}$, the hypervolume indicator $\mathcal{H}$ of this front $\mathcal{P}$ w.r.t. a reference point $\mathbf{r}$ is defined as follows [53, 54]:

$$\mathcal{H}(\mathcal{P}, \mathbf{r}) = \lambda_M \left(\cup_{\mathbf{y} \in \mathcal{P}} [\mathbf{r}, \mathbf{y}] \right) \tag{8}$$

where λ_M is the Lebesgue measure of the region that dominates $\mathbf{r}$ and that is dominated by $\mathcal{P}$ (in $\mathbb{R}^M$, i.e., for $\mathbb{R}^2$, λ_2 is the area of the dominated region, while on $\mathbb{R}^3$, λ_3 is the volume).

Using this definition, we can define the *hypervolume improvement* (also referred to as exclusive hypervolume) generated by a new point $\mathbf{y}_*$ as:

$$\mathcal{H}_{exc}(\mathbf{y}_*, \mathcal{P}, \mathbf{r}) = \mathcal{H}(\mathcal{P} \cup \{\mathbf{y}_*\}, \mathbf{r}) - \mathcal{H}(\mathcal{P}, \mathbf{r}) \tag{9}$$

.

Based on the definition of hypervolume improvement in equation 9, we can define the Expected HyperVolume Improvement (EHVI) at an arbitrary new design point $\mathbf{x}_\star$ as:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) = \int_{\mathbb{R}^M} \mathcal{H}_{exc}(\mathbf{y}, \mathcal{P}, \mathbf{r}) \cdot \boldsymbol{\xi}_{\sigma,\mu}(\mathbf{y}) d\mathbf{y} \tag{10}$$

where $\mathbf{y}$ corresponds to a (random) M-variate objective vector, while $\boldsymbol{\xi}_{\sigma,\mu}(\mathbf{y})$ denotes the value of the M-variate independent normal density function in this vector (given the predictive mean vector $\boldsymbol{\mu} \in \mathbb{R}^M$ at $\mathbf{x}_\star$, and the predictive variance vector $\boldsymbol{\sigma}^{\mathbf{2}} \in \mathbb{R}^M_+$ at $\mathbf{x}_\star$). For ease of notation, let $\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, \mathcal{P}, \mathbf{r}) := \text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma})$.

Let us define a set $\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r})$ which contains (given a Pareto front $\mathcal{P}$, the reference point $\mathbf{r}$ and the output vector $\mathbf{y}$) all the output vectors that currently do *not* belong to the dominated set, but that would be added to it when the vector $\mathbf{y}$ were added to the front.[33, 55]:

$$\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r}) = \left\{ \mathbf{z} \in \mathbb{R}^M \mid \mathbf{y} \prec \mathbf{z} \ , \ \mathbf{z} \prec \mathbf{r} \ \text{ and } \nexists \mathbf{q} \in \mathcal{P} : \mathbf{q} \prec \mathbf{z} \right\} \tag{11}$$

For notational simplicity, let $\Delta(\mathbf{y}, \mathcal{P}, \mathbf{r}) := \Delta(\mathbf{y})$. The EHVI in equation 10 can then be rewritten as:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{N_M} \left(\int_{-\infty}^{y_1=u_1^{(i)}} \cdots \int_{-\infty}^{y_M=u_M^{(i)}} \right) \lambda_M \left[S_M^{(i)} \cap \Delta(\mathbf{y}) \right] \cdot \xi_{\mu,\sigma}(\mathbf{y}) d\mathbf{y} \tag{12}$$

where λ_M refers to the M-dimensional Lebesgue measure, $S_M^{(i)}$ refers to hyperbox i (see equation B20 in Appendix B.3), and N_M denotes the total number of hyperboxes in the decomposition. Note that equation 12 allows for piece-wise integration, given the summation over the different hyperboxes.

Dividing each integration slice $\int_{-\infty}^{y_m=u_m^{(i)}}$ into $(\int_{-\infty}^{y_m=l_m^{(i)}} + \int_{l_m^{(i)}}^{y_m=u_m^{(i)}})$, we obtain:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{N_M} \left(\left(\int_{-\infty}^{y_1=l_1^{(i)}} + \int_{y_1=l_1^{(i)}}^{y_1=u_1^{(i)}} \right) \cdots \left(\int_{-\infty}^{y_M=l_M^{(i)}} + \int_{y_M=l_M^{(i)}}^{y_M=u_M^{(i)}} \right) \right) \lambda_M \left[S_M^{(i)} \cap \Delta(\mathbf{y}) \right] \cdot \boldsymbol{\xi}_{\mu,\sigma}(\mathbf{y}) d\mathbf{y} \tag{13}$$

As evident, each of the individual terms of this sum consist of the multiplication of M factors, each of which contains the sum of 2 integrals. Since integration is a linear mapping, we can expand each individual term in equation 13, resulting in a summation of 2^M terms, each consisting of an M-dimensional integral.

Let us finally define $\mathbf{C}^{(j)_2}$ as a binary representation of such an M-dimensional integral. $\mathbf{C}^{(j)_2}$'s length is thus equal to M. The kth element, $C_k^{(j)_2}$, equals 0 if the kth integral has finite bounds, and 1 if the lower bound is $-\infty$.

Using the results from [26, 33], the EHVI can then be calculated exactly as follows:

$$\text{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{N_M} \left(\sum_{j=0}^{2^M} \left(\prod_{k=1}^{M} \omega_e(i, k, C_k^{(j)_2}) \right) \right) \tag{14}$$

with

$$\omega_e \left(i, k, C_k^{(j)_2} \right) := \begin{cases} \Psi \left(l_k^{(i)}, u_k^{(i)}, \mu_k, \sigma_k \right) & \text{if } C_k^{(j)_2} = 0 \\ \vartheta \left(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k \right) & \text{if } C_k^{(j)_2} = 1 \end{cases} \tag{15}$$

where i refers to the index number of the hyperbox, k to the index number of the objective function, and $C_k^{(j)_2}$ is the binary representation of the kth objective. Note that equation 14 implicitly uses the independence assumption

between the different objectives, to replace the M-dimensional integrals by multiplication of M single-dimensional integrals. As evident from equation 15, the calculation of these single integrals can be done exactly, but depends on whether the integral has finite bounds ($C_k^{(j)_2} = 0$) or an infinite lower bound ($C_k^{(j)_2} = 1$). More specifically, for $C_k^{(j)_2} = 1$, we have:

$$\vartheta\left(l_k^{(i)}, u_k^{(i)}, \sigma_k, \mu_k\right) = \left(u_k^{(i)} - l_k^{(i)}\right) \cdot \left(\Phi\left(\frac{l_k^{(i)} - \mu_k}{\sigma_k}\right)\right) \tag{16}$$

where Φ denotes the Cumulative Distribution Function (CDF) of the standard normal distribution. Equation 16 also occurs in [33], but we adjust it here for a minimization context.

For $C_k^{(j)_2} = 0$, we have:

$$\begin{aligned}\Psi\left(l_k^{(i)}, u_k^{(i)}, \mu_k, \sigma_k\right) = \Psi_{-\infty}\left(u_k^{(i)}, u_k^{(i)}, \mu_k, \sigma_k\right) - \\ \Psi_{-\infty}\left(u_k^{(i)}, l_k^{(i)}, \mu_k, \sigma_k\right)\end{aligned} \tag{17}$$

with

$$\Psi_{-\infty}(a, b, \mu, \sigma) := \int_{-\infty}^{b} (a - z)\frac{1}{\sigma}\phi\left(\frac{z - \mu}{\sigma}\right) dz \tag{18}$$

$$= \sigma\phi\left(\frac{b - \mu}{\sigma}\right) + (a - \mu)\left[\Phi\left(\frac{b - \mu}{\sigma}\right)\right] \tag{19}$$

where ϕ and Φ denote the Probability Density Function (PDF) and Cumulative Distribution Function (CDF) of the standard normal distribution, respectively.

We can further refine equation 14 to deal efficiently with cheap and expensive objectives. To that end, we introduce an M-dimensional binary vector $\mathbf{t}^f$: the kth element of this vector, t_k^f, equals 0 if the kth objective is cheap, and 1 otherwise.We can then efficiently calculate the resulting Cheap-Expensive Hypervolume Improvement (CEHVI) as follows:

$$\text{CEHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{i=1}^{N_M}\left(\sum_{j=0}^{2^M}\left(\prod_{k=1}^{M}\omega(i, k, C_k^{(j)_2}, t_k^f)\right)\right) \tag{20}$$

with

$$\omega(i, k, C_k^{(j)_2}, t_k^f) := \begin{cases}\omega_c(i, k, C_k^{(j)_2}) & \text{if } t_k^f = 0 \\ \omega_e(i, k, C_k^{(j)_2}) & \text{if } t_k^f = 1\end{cases} \tag{21}$$

For the expensive objectives, this expression reduces to ω_e as given by equation 15. For the cheap objectives, the calculation of $\omega_c(i, k, C_k^{(j)_2})$ depends on the relative location of y_k w.r.t. $l_k^{(i)}$ and $u_k^{(i)}$, as shown in Table 4. As evident, the resulting values are *deterministic*.

Table 4: Calculation of $\omega_c(i, k, C_k^{(j)_2})$ for the cheap objectives

Condition	Value
$y_k < l_k^{(i)} < u_k^{(i)}$	$\omega_c\left(i, k, C_k^{(j)_2}\right) = \begin{cases} 0 & \text{if } C_k^{(j)_2} = 0 \\ (u_k^{(i)} - l_k^{(i)}) & \text{if } C_k^{(j)_2} = 1 \end{cases}$
$l_k^{(i)} < y_k < u_k^{(i)}$	$\omega_c\left(i, k, C_k^{(j)_2}\right) = \begin{cases} (u_k^{(i)} - y_k) & \text{if } C_k^{(j)_2} = 0 \\ 0 & \text{if } C_k^{(j)_2} = 1 \end{cases}$
$l_k^{(i)} < u_k^{(i)} < y_k$	$\omega_c\left(i, k, C_k^{(j)_2}\right) = 0$

4.2.2 Acquisition Function Maximization

In every iteration, the query points are obtained by maximizing the acquisition function (usually referred to as *inner optimization*). We use a multi-start optimization method [56, 57] that incorporates the cheap constraints. First, the Monte Carlo method is used to sample 5000 points, and the points that violate cheap constraints are removed. We then select the 10 points with the highest CEHVI-C value and apply Sequential Least Square Programming (SLSQP) with the cheap constraints [58] to each of these in parallel (using the SLSQP implementation of the Scipy [59] library).

5 Result and Discussion

5.1 Experiment settings

The proposed hypervolume-based MOBGO algorithm was implemented using Trieste [60] in Python. Before applying our method to the Motor Optimization case, we consider five benchmark functions to test the performance of the proposed algorithm, by testing it on three unconstrained optimization problems (DTLZ1, DTLZ2, and DTLZ3 [61]) and two constrained optimization problems (BNH [62] and SRN [63]). The characteristics of these benchmark functions are presented in Table 5. The reference point indicated in the table is used for the hypervolume computations.

For these benchmark functions, $(d \times 11 + 1)$ initial design points were generated using quasi-random Halton Sampling [64]. The proposed method is compared with EHVI(-Constrained), Random sampling, and NSGA-II (see [65]; we used the version present in PyMOO [66], which accounts for constraints). The total budget for the FEM simulator is set to 100 input evaluations, except for the NSGA-II algorithm: as this method is less data efficient,

Table 5: Full specification of the benchmark functions.

Name	Input d	Objectives		Constraints		Reference point
		Expensive	Cheap	Expensive	Cheap	
DTLZ1	6	2	1	0	0	[425, 425, 425]
DTLZ2	6	2	1	0	0	[2.5 , 2.5 , 2.5]
DTLZ3	6	2	1	0	0	[825, 825, 825]
BNH	2	1	1	1	1	[150, 100]
SRN	2	1	1	1	1	[800, 200]

we allow it to spend 250 input evaluations. The AL budget is set to zero, as the expensive constraints (if any) are not hard to model. Evidently, for the unconstrained problems, the CEHVI-C acquisition function reduces to the CEHVI acquisition function (see equation 7).

For the PMSM optimization problem, 35 initial points are generated using Latin Hypercube sampling [52]. The AL budget for the BO methods is set to 10 iterations; we include the AL phase in this problem as the initial design is small, so we expect it to be beneficial, especially for learning the hard-to-model average torque (Tavg) constraint. Both the initial design and the number of AL iterations are deliberately kept small as the FEM model is relatively slow to run. We compare the performance of our algorithm against the same competitors as in the benchmark functions. The total budget equals 100 input evaluations, except for the NSGA-II algorithm (250 evaluations).

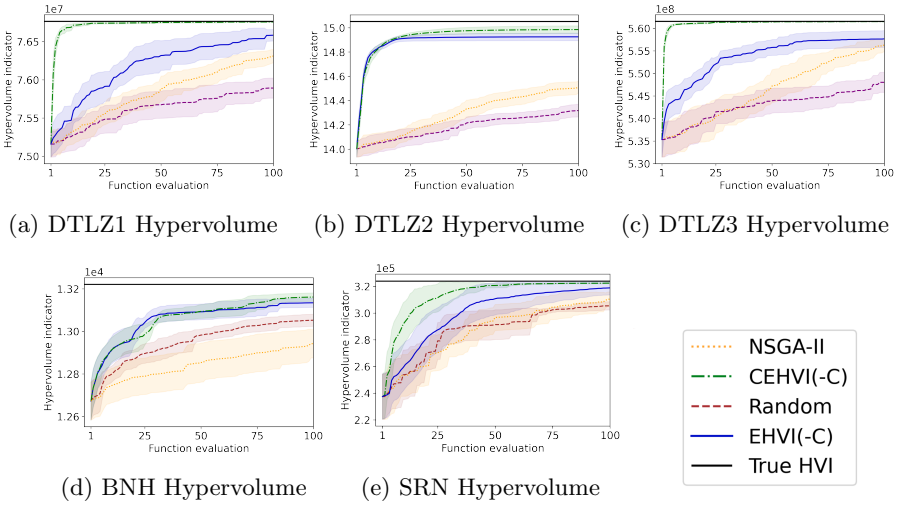
5.2 Results for benchmark functions

We carried out 10 repetitions for each of the benchmark experiments, to check the robustness of the results against the randomness involved in the algorithms (which is evident in the NSGA-II and Random algorithms; in the BO algorithms, it impacts the multistart design of the inner optimization).

Figure 3 shows the evolution of the mean hypervolume on the different benchmark functions, for the competing algorithms. As shown, the BO approaches (CEHVI and EHVI) clearly outperform the competing algorithms in the unconstrained problem settings (top row). Moreover, the CEHVI algorithm has significantly better performance than the EHVI algorithm in the DTLZ1 and DTLZ3 experiments, which have a disjoint Pareto front [61] and are thus hard to optimize (the DTLZ2, by contrast, has a smooth Pareto front).

In the constrained benchmark problems (bottom row), CEHVI-C again has a clearly higher hypervolume indicator value than the other methods. While both problems have a smooth Pareto front, CEHVI-C outperforms EHVI-C in the SRN problem (in the BNH problem, the performance of both algorithms is similar, as the cheap function is smoother and thus relatively easy to model with GPs).

Table 6 gives an overview of the final hypervolume obtained at the end of the different algorithms, along with the difference (in %) from the true optimal hypervolume. As evident from this table, CEHVI-C is the winner in



(a) DTLZ1 Hypervolume (b) DTLZ2 Hypervolume (c) DTLZ3 Hypervolume

(d) BNH Hypervolume (e) SRN Hypervolume

Fig. 3: Evolution of the mean hypervolume on the benchmark problems, with a total budget of 100 evaluations. The shaded areas reflect the 95% confidence intervals based on 10 repetitions.

all test problems except in BNH. Note, though, that NSGA-II only succeeds in outperforming both BO approaches here because we gave it a significantly higher total budget; for limited budgets ($<= 100$), NSGA-II is clearly inferior (as evident from figure 3).

5.3 Results for PMSM design problem

To check the robustness of the algorithms against randomness, we ran 10 repetitions. The hypervolume values are calculated with reference point [Efficiency, Torque ripple, Total mass] = $[0.80, 8.1, 23.]$.

Figure 4 shows the evolution of the mean hypervolume for the different algorithms. For the BO algorithms, we used the first ten iterations to implement an AL stage: here, points were queried by implementing the FPV acquisition function on the average torque constraint, to improve the corresponding GP model. As evident from the figure, the AL phase already succeeds in improving the hypervolume. In the optimization stage, we see that CEHVI-C performs better than EHVI-C; NSGA-II is clearly inferior to both BO approaches. Actually, it even fails to query feasible points at certain iterations (even the later ones) due to the many constraints in the PMSM design problem. As a result, its hypervolume only improves marginally in the first 100 iterations. As evident from table 7, which shows the expected hypervolume obtained at the end of the algorithms, the improvements obtained remain marginal even at a higher budget.

Table 6: Overview of the mean hypervolume and the % difference from the true optimal hypervolume (with 95% confidence interval), obtained at the end of the algorithms, for the benchmark problems. The best result for each problem is highlighted in bold.

Test Problem	BO Budget	Method	HVI	Δ to Ground truth (%)
DTLZ1	100	Random	7.5892e7 ± 1.3126e5	1.137 71 ± 0.275 86
	100	EHVI	7.6585e7 ± 7.5134e4	0.235 31 ± 0.157 91
	100	CEHVI	**7.6759e7** ± 4.6615e3	**0.008 22** ± **0.009 79**
	250	NSGA2	7.6571e7 ± 6.8405e4	0.254 02 ± 0.143 76
DTLZ2	100	Random	1.4318e1 ± 0.052 56	4.868 01 ± 0.563 46
	100	EHVI	1.4925e1 ± 0.031 35	0.835 12 ± 0.336 07
	100	CEHVI	**1.4985e1** ± 0.031 13	**0.438 29** ± **0.333 69**
	250	NSGA2	1.4713e1 ± 0.039 97	2.246 03 ± 0.428 48
DTLZ3	100	Random	5.4802e8 ± 2.2293e6	2.404 30 ± 0.640 53
	100	EHVI	5.5766e8 ± 1.8455e6	0.686 36 ± 0.530 25
	100	CEHVI	**5.6148e8** ± 2.5549e4	**0.006 21** ± **0.007 34**
	250	NSGA2	5.5988e8 ± 5.3153e5	0.291 95 ± 0.152 72
BNH	100	Random	1.3127e4 ± 1.6295e1	0.699 74 ± 0.198 86
	100	EHVI-C	1.3134e4 ± 2.1248e1	0.651 31 ± 0.259 31
	100	CEHVI-C	1.3161e4 ± 1.9731e1	0.447 41 ± 0.240 80
	250	NSGA2	**1.3174e4** ± 3.2091e1	**0.350 01** ± **0.391 63**
SRN	100	Random	3.1101e5 ± 2.9123e3	3.958 08 ± 1.451 03
	100	EHVI-C	3.1885e5 ± 5.2265e3	1.536 36 ± 2.604 01
	100	CEHVI-C	**3.2240e5** ± 4.2819e2	**0.439 08** ± **0.213 33**
	250	NSGA2	3.1885e5 ± 1.9090e3	1.536 84 ± 0.951 11

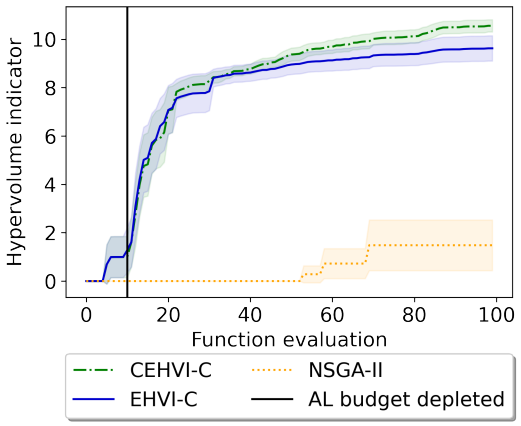


Fig. 4: Evolution of the mean hypervolume for the PMSM design problem, for a total budget of 100 iterations. The shaded areas reflect the 95% confidence interval based on 10 repetitions.

Figure 5 illustrates the quality of the final Pareto front obtained by the CEHVI-C and the EHVI-C methods for a single arbitrary run. Clearly, CEHVI-C succeeds in achieving solutions with a lower Torque ripple and a

Table 7: Overview of the mean hypervolume (with 95% confidence interval halfwidth), obtained at the end of the algorithms, for the PMSM design problem. The best result is highlighted in bold.

Method	AL Budget	Total Budget	HVI
EHVI-C	10	100	9.6239 ± 0.2607
CEHVI-C	10	100	**10.5554** ± 0.1258
NSGA-II	0	250	1.4818 ± 0.5237

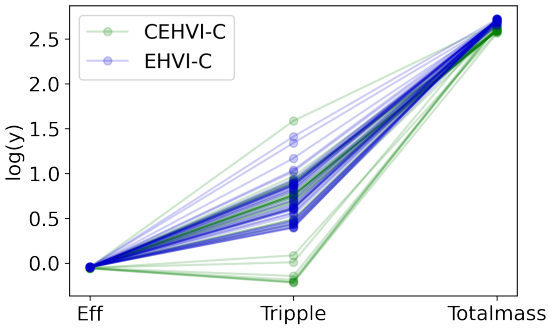


Fig. 5: Log of Pareto front of a single experiment run.

lower Total mass than EHVI-C without compromising motor efficiency. The CEHVI-C runs also take less computation time, as it avoids any estimations for the cheap objective and constraints.

The three objectives of the optimization are the torque ripple, the motor efficiency, and the total mass. The magnet mass is also added because it is the most expensive part of the machine. Generally speaking, the cost of 1 kg of rare-earth magnets equals more than ten times that of 1 kg of copper or 1 kg of iron [67, 68].

Figure 6 shows the optimal geometry of an (arbitrary) Pareto-optimal PMSM design obtained by CEHVI-C, along with the flux lines and flux densities. While there is saturation in some parts of the core, it does not impact the performance metrics in any negative way.

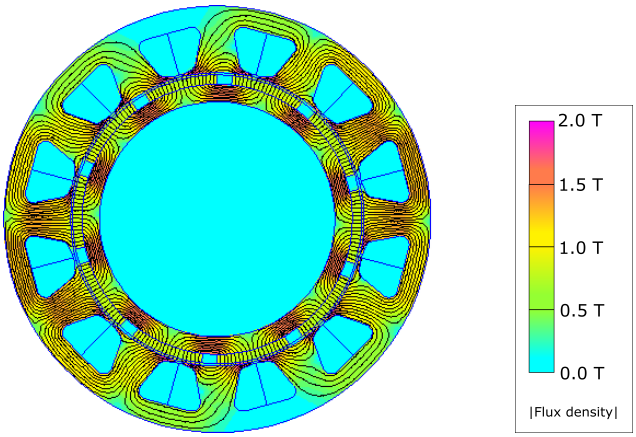


Fig. 6: One of the Pareto-optimal designs found by CEHVI-C.

6 Conclusion

In this paper, a hypervolume-based MOBGO approach has been presented and applied in view of optimizing a Permanent Magnet Synchronous Motor design. This design problem consists of a mix of expensive performance metrics (which require FEM evaluations) and cheap performance metrics (which can be evaluated using closed-form expressions). The key strength of the proposed approach is that it distinguishes between these cheap and expensive functions, by only estimating Gaussian Process models for the expensive outcomes. It includes an active learning stage (which uses the FPV acquisition function to improve the accuracy for hard-to-model constraints) and an optimization phase (which uses the proposed CEHVI-C acquisition function, which is a constrained and cheap-expensive version of the well-known EHVI criterion). The performance of the CEHVI-C function was first evaluated on a number of benchmark functions; as shown, it leads to superior performance over the standard EHVI-based approaches, especially when the cheap objective(s) are hard to model with GP. This superiority was further confirmed in the PMSM design results.

The proposed approach is likely beneficial for other engineering design problems that include cheap and expensive outcomes. In future research, we plan to extend the approach further such that it can handle noisy function evaluations. Another interesting topic is to extend the method to be cost-aware. Indeed, the cost of an expensive function evaluation may not be the same over the entire search space; cost-aware BO may then try to find the optimal solutions while minimizing both the number of function evaluations and the resulting evaluation cost.

Acknowledgments. This work has been supported by the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" and the "Fonds Wetenschappelijk Onderzoek (FWO)" programmes. We also thank our colleague Jixiang Qing for providing detailed comments that helped improve the paper.

Appendix A Details of the PMSM Problem

A.1 Expensive Objectives

The expensive objectives are evaluated using Finite Element Methods (FEM) [37]. The motor torque objective is given by:

$$T_m = \frac{p}{2} \frac{\pi \mu_o D_g L_{fe}}{2 L_g} F_s F_r \sin(\delta_{sr}) \tag{A1}$$

where p refers to the number of rotor poles, μ_o is the permeability of the air, D_g is the airgap diameter, L_{fe} is the axial length, and L_g is the length of the airgap. The notations F_s, F_r, and δ_{sr} refer to the magnetomotive force of the stator and the rotor, and the angle between them, respectively. F_s depends mainly on the winding geometry (area, number of turns, and phases) and on the permeable current density. F_r depends mainly on the properties and geometry of the magnet, as well as on g and L_{fe}. The calculation of F_s and F_r relies on the expensive FEM evaluation.

The efficiency of the motor is given by:

$$\eta = \frac{T_m \omega_r}{T_m \omega_r + P_l} 100 \tag{A2}$$

where ω_r is the rotor speed, and P_l refers to the motor losses: copper loss, magnet loss, and iron (stator and rotor core losses). P_l depends on the flux density, geometry, material properties, current density, and the speed of the motor. P_l is evaluated by FEM.

A.2 Cheap objectives and constraints

A.2.1 Total mass calculation

The cheap objective function for the motor design problem is the total mass of the following motor parts: part_1 = stator core (silicon steel), part_2 = rotor core (silicon steel), part_3 = winding (copper), and part_4 = rotor poles (magnets). The mass of part_n can be calculated as follows:

$$\text{Volume part}_n = \text{Area part}_n \times \text{Length part}_n \tag{A3}$$

$$\text{Mass part}_n = \text{Volume part}_n \times \text{Density part}_n \tag{A4}$$

where steel density = 7267.5 kg/m^2, copper density = 8933 kg/m^2, magnet density = 7400 kg/m^2. Then, the total mass of the motor can be obtained as follows:

$$\text{Total mass} = \sum_{n=1}^{4} \text{Mass part}_n \tag{A5}$$

A.2.2 Cheap constraints

For the calculation of the cheap constraint functions, we first need to define the following constants: airgap length (L_g) = 1, slot opening height (soh) = 1, slot wedge width (swx) = 1, slot width yoke side ratio (swyR) = 1.5, slots = 12. The stator Y thickness (SYt) can then be calculated using following formulas:

$$\text{Rslotmiddle} = \text{ROR} + L_g + \text{soh} + \text{swx} + (0.5 \times \text{sh}) \tag{A6}$$

$$\text{LMslot} = \text{STR} \times 2\pi \times \frac{\text{Rslotmiddle}}{\text{slots}} \tag{A7}$$

$$\text{sw} = \text{swyR} \times \text{LMslot} \tag{A8}$$

$$\text{Rteeth} = \sqrt{(\text{ROR} + L_g + \text{soh} + \text{swx} + \text{sh})^2 + (0.5 \times \text{sw})^2} \tag{A9}$$

$$\text{SYt} = \text{SOR} - \text{Rteeth} \tag{A10}$$

The second cheap constraint, shaft diameter (ShaftD) is defined by:

$$\text{ShatfD} = 2 \times \text{ROR} - 2 \times \text{RYt} - 2 \times \text{Thpm} \tag{A11}$$

Appendix B Background on Bayesian Optimization

B.1 GP model Details

The Gaussian Process (GP) model [13, 14, 43] is the most popular type of surrogate model used in BO, especially if the input domain is continuous. Informally, a GP defines a distribution over real-valued functions: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, and is fully specified by its mean function $m(\mathbf{x})$ and its (positive semi-definite) covariance function $k(\mathbf{x}, \mathbf{x}')$.

A GP provides a predictive distribution for the output function under study at unobserved input locations in the search space, given a (limited) set of available input/output data. Suppose we want to model an output function f_m, for which we have evaluated the set of data points $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, yielding function evaluations $Y_m = [f_m(\mathbf{x}_1), \ldots, f_m(\mathbf{x}_N)]$. Then, $\mathcal{D}_N = \{X, Y_m\}$ is defined as the observed data so far in our optimization process, and the GP model is trained on these data (usually by means of maximum likelihood estimation, as discussed below).

The means and variances of the predictive distribution $\boldsymbol{f}_\star$, at any set of unobserved data points $X_\star = [\mathbf{x}_{\star 1}, \ldots, \mathbf{x}_{\star L}]$, can then be estimated as follows:

$$\mu_m(X_\star) = \mathbb{E}(\boldsymbol{f}_\star \mid X_\star, \mathcal{D}_N) = K_{\star \mathrm{x}} K_{\mathrm{xx}}^{-1} Y_m \tag{B12}$$

$$\sigma_m^2(X_\star) = \mathrm{Var}(\boldsymbol{f}_\star \mid X_\star, \mathcal{D}_N) = K_{\star\star} - K_{\star \mathrm{x}} K_{\mathrm{xx}}^{-1} K_{\star \mathrm{x}}^T \tag{B13}$$

where $\mu_m(X_\star)$ is the $L \times 1$ vector with the predictive means, and $\sigma_m^2(X_\star)$ is the $L \times 1$ vector with the predictive variances. The notation K_{xx} refers to the $N \times N$ matrix containing the estimated covariances between the available data, i.e., $k(\mathbf{x}_i, \mathbf{x}_j)$ for $i, j = 1 \ldots N$. The notation $K_{\star \mathrm{x}}$ is the $L \times N$ matrix containing the covariance estimates between the new points $X_\star$ and the N available points, i.e., $k(\mathbf{x}_{\star i}, \mathbf{x}_j)$, for $i = 1 \ldots L, j = 1 \ldots N$. The notation $K_{\star\star}$ refers to the $L \times L$ matrix containing the estimated covariances between the new points, i.e., $k(\mathbf{x}_{* i}, \mathbf{x}_{* j})$ for $i, j = 1 \ldots L$. To model the covariance function, we choose the Matérn 5/2 kernel [69]. This kernel is a common choice when the smoothness of the function is unknown [6], since it does not make any overly smooth assumptions with respect to the output function under study. It is defined as follows:

$$k(\mathbf{x}, \mathbf{x}') = \alpha \left(1 + \sqrt{5} r + \frac{5}{3} r^2\right) \exp(-\sqrt{5} r), \tag{B14}$$

$$r = \sqrt{\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{l_i^2}} \tag{B15}$$

where α is the kernel variance, and l_i is the kernel length scale for the ith dimension.

When training the GP, Maximum Likelihood Estimation (MLE) [70] is commonly used to estimate the hyperparameters $\theta := \{\alpha, l_1, \ldots, l_d\}$:

$$\hat{\theta} = \arg\max_\theta \log p(\boldsymbol{f} \mid X, \theta) \tag{B16}$$

$$= \arg\max_\theta -\frac{1}{2}(\log|2\pi K_{\mathrm{xx}}| + \boldsymbol{f}^T K_{\mathrm{xx}}^{-1} \boldsymbol{f}) \tag{B17}$$

In the MOBGO case, each expensive output function (objectives as well as constraints) is modeled using a distinct, *single-output* GP.

B.2 Expected Improvement

In unconstrained single-objective optimization problems, one of the most popular acquisition functions is the Expected Improvement (EI) [1]. As evident from its name, it measures the *improvement* in the objective outcome that

the analyst may expect when querying a new point $\mathbf{x}_*$, given the current best objective outcome obtained so far ($\hat{y}$) and the current GP model for the objective function (estimated on the currently available data $\mathcal{D}_n$). Given the GP model assumptions, the predictive outcome $f(\mathbf{x}_*)$ at such a new point is normally distributed ($N(\mu, \sigma^2)$, with μ the predictive mean at $\mathbf{x}_*$ and σ^2 the predictive variance). The improvement function at any arbitrary new point $\mathbf{x}_*$ is then given by the following random variable (without loss of generality, we assume here that we aim to *minimize* the objective function):

$$I(\mathbf{x}_*) := (\hat{y} - f(\mathbf{x}_*))\mathbb{I}(\hat{y} > f(\mathbf{x}_*)) \tag{B18}$$

where $\mathbb{I}$ is the indicator function. The EI at $\mathbf{x}_*$ is given by the following closed form expression:

$$\begin{aligned} \mathrm{EI}\left(\mathbf{x}_*;\ \mathcal{D}_n\right) :=& \mathbb{E}[I(\mathbf{x}_*)] \\ =& \sigma\phi\left(\frac{\hat{y}-\mu}{\sigma}\right) + (\hat{y}-\mu)\,\Phi\left(\frac{\hat{y}-\mu}{\sigma}\right) \end{aligned} \tag{B19}$$

B.3 Hyperbox Decomposition

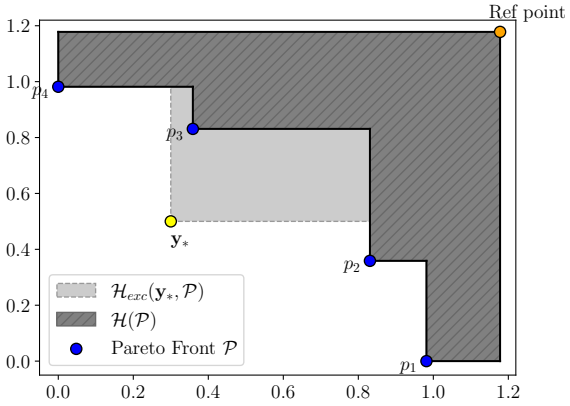


Fig. B1: Illustration of the hypervolume improvement (light grey area) of a new point $\mathbf{y}_*$ given the Pareto front $\mathcal{P}$.

The concept of hypervolume improvement ($\mathcal{H}_{exc}$) in $\mathbb{R}^2$ is illustrated in Figure B1. To calculate $\mathcal{H}_{exc}$ efficiently (using piece-wise integration), the non-dominated space is partitioned into a set of hyper-boxes or hyper-cells (as few boxes/cells as possible).

Figure B2 illustrates this hyper-box decomposition in $\mathbb{R}^2$ for a Pareto front $\mathcal{P}$ consisting of 4 points. Note that each hyperbox $S_2^{(i)}$ $(i = 1, \cdots, 5)$ in this figure can be represented by its lower bound vector $\mathbf{l}_2^{(i)}$ and its upper bound vector $\mathbf{u}_2^{(i)}$ (both vectors are 2-dimensional in this case). In $\mathbb{R}^M$, the hyper-boxes are thus represented by:

$$S_M^{(i)} = \left(\mathbf{l}_M^{(i)}, \mathbf{u}_M^{(i)}\right) = \left(\left(l_1^{(i)}, \ldots, l_M^{(i)}\right)^\top, \left(u_1^{(i)}, \ldots, u_M^{(i)}\right)^\top\right) \quad \text{for} \quad i = 1, \ldots, N_M \tag{B20}$$

where N_M is the number of hyper-boxes.

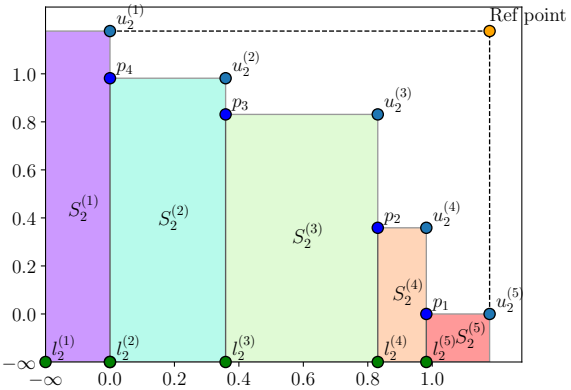


Fig. B2: Hyperbox decomposition on $\mathbb{R}^2$; each hyperbox is shown in a different color.

Different box-partition algorithms have been presented in the literature, see for instance [25, 26, 33, 71, 72]. In this paper, we use the box-partition algorithm from [25]. This is without loss of generality since the proposed algorithm in this paper is compatible with *any* box-partition algorithm.

Appendix C DTLZ results with varying input and output dimensions

To assess the performance of our proposed method under varying input and output dimensions, we conducted an evaluation of the DTLZ functions with combinations of input dimensions of [4, 6, 8] and output dimensions of [2, 3]. All of the scenarios assume 1 cheap objective. The obtained results for DTLZ1, DTLZ2, and DTLZ3 are presented in Tables C1, C2, and C3, respectively.

Table C1: Hypervolume improvement with 95% confidence interval for the DTLZ1 function. Reference points of [600, 600] and [600, 600, 600] are used for M=2 and M=3 respectively.

M	d	Method	Budget	HVI	Δ to Ground truth (%)
2	4	Random	100	3.5801e5 ± 3.8568e2	0.550 50 ± 0.172 85
		EHVI	100	3.5941e5 ± 1.7502e2	0.164 43 ± 0.078 44
		CEHVI	100	**3.5995e5** ± 4.0028e1	**0.014 62** ± 0.017 94
		NSGA2	200	3.5973e5 ± 8.0121e1	0.074 92 ± 0.035 91
	6	Random	100	3.5127e5 ± 1.4763e3	2.424 10 ± 0.661 64
		EHVI	100	3.5694e5 ± 1.0928e3	0.850 69 ± 0.489 78
		CEHVI	100	**3.5976e5** ± 8.4896e1	**0.066 34** ± 0.038 04
		NSGA2	200	3.5780e5 ± 5.3252e2	0.611 80 ± 0.238 66
	8	Random	100	3.3646e5 ± 2.4519e3	6.538 40 ± 1.098 90
		EHVI	100	3.5194e5 ± 1.5345e3	2.237 70 ± 0.687 71
		CEHVI	100	**3.5931e5** ± 1.4983e2	**0.192 46** ± 0.067 15
		NSGA2	200	3.5162e5 ± 1.5136e3	2.327 00 ± 0.678 34
3	4	Random	100	2.1574e8 ± 6.3676e4	0.122 57 ± 0.047 56
		EHVI	100	2.1598e8 ± 1.1133e4	0.010 97 ± 0.008 32
		CEHVI	100	**2.160 08** ± 8.4567e2	**0.000 36** ± 0.000 63
		NSGA2	200	2.1598e8 ± 8.8661e3	0.008 63 ± 0.006 62
	8	Random	100	2.1597e8 ± 3.4617e4	0.015 09 ± 0.025 86
		EHVI	100	2.1550e8 ± 2.3715e5	0.229 61 ± 0.177 14
		CEHVI	100	**2.1597e8** ± 3.4617e4	**0.015 09** ± 0.025 86
		NSGA2	200	2.1475e8 ± 2.0456e5	0.580 44 ± 0.152 79

Table C2: Hypervolume improvement with 95% confidence interval of the DTLZ2. Reference points of [3, 3] and [3, 3, 3] are used for M=2 and M=3 respectively.

M	d	Method	Budget	HVI	Δ to Ground truth (%)
2	4	Random	100	0.8046e1 ± 0.0015e1	1.996 80 ± 0.295 09
		EHVI	100	0.8183e1 ± 0.0001e1	0.319 22 ± 0.021 89
		CEHVI	100	**0.8192e1** ± 0.0001e1	**0.217 39** ± 0.024 81
		NSGA2	200	0.8152e1 ± 0.0011e1	0.701 69 ± 0.226 46
	6	Random	100	0.7895e1 ± 0.0021e1	3.837 50 ± 0.424 09
		EHVI	100	0.8171e1 ± 0.0001e1	0.469 86 ± 0.026 71
		CEHVI	100	**0.8191e1** ± 0.0001e1	**0.228 89** ± 0.020 07
		NSGA2	200	0.8083e1 ± 0.0011e1	1.553 90 ± 0.212 20
	8	Random	100	0.7709e1 ± 0.0025e1	6.110 00 ± 0.500 78
		EHVI	100	0.8160e1 ± 0.0003e1	0.609 07 ± 0.077 58
		CEHVI	100	**0.8180e1** ± 0.0002e1	**0.362 47** ± 0.049 17
		NSGA2	200	0.7950e1 ± 0.0021e1	3.173 40 ± 0.410 99
3	4	Random	100	2.5836e1 ± 4.2026e2	2.232 70 ± 0.256 59
		EHVI	100	2.6318e1 ± 2.5326e2	0.409 31 ± 0.154 60
		CEHVI	100	**2.6345e1** ± 3.0863e2	**0.307 18** ± 0.188 43
		NSGA2	200	2.6188e1 ± 4.5699e2	0.898 02 ± 0.279 01
	8	Random	100	2.5252e1 ± 0.0072e1	4.439 90 ± 0.437 50
		EHVI	100	2.6251e1 ± 0.0063e1	0.661 86 ± 0.381 74
		CEHVI	100	**2.6260e1** ± 0.0114e1	**0.627 20** ± 0.697 41
		NSGA2	200	2.5836e1 ± 0.0081e1	2.230 50 ± 0.495 00

Table C3: Hypervolume improvement with 95% confidence interval of the DTLZ3. Reference points of [1100, 1100] and [1100, 1100, 1100] are used for M=2 and M=3 respectively.

M	d	Method	Budget	HVI	Δ to Ground truth (%)
2	4	Random	100	1.1989e6 ± 2.1987e3	0.918 26 ± 0.293 17
		EHVI	100	1.2068e6 ± 1.1462e3	0.266 29 ± 0.152 83
		CEHVI	100	**1.2098e6** ± 2.3727e2	**0.013 97** ± 0.031 63
		NSGA2	200	1.2082e6 ± 4.9900e2	0.150 86 ± 0.066 54
	6	Random	100	1.1611e6 ± 9.0820e3	4.041 20 ± 1.211 00
		EHVI	100	1.1952e6 ± 4.9118e3	1.220 10 ± 0.654 93
		CEHVI	100	**1.2090e6** ± 3.1147e2	**0.084 70** ± 0.041 53
		NSGA2	200	1.2005e6 ± 3.2024e3	0.788 01 ± 0.427 00
	8	Random	100	1.0779e6 ± 1.4485e4	10.916 00 ± 1.931 40
		EHVI	100	1.1688e6 ± 1.2016e4	3.408 90 ± 1.602 20
		CEHVI	100	**1.2061e6** ± 1.0296e3	**0.321 77** ± 0.137 28
		NSGA2	200	1.1701e6 ± 7.6555e3	3.301 00 ± 1.020 80
3	4	Random	100	1.3275e9 ± 8.7369e5	0.265 79 ± 0.105 91
		EHVI	100	1.3306e9 ± 5.7116e5	0.033 40 ± 0.069 23
		CEHVI	100	**1.3310e9** ± 3.6666e3	**0.000 18** ± 0.000 44
		NSGA2	200	1.3308e9 ± 1.0310e5	0.014 52 ± 0.012 50
	8	Random	100	1.2705e9 ± 7.1915e6	4.548 70 ± 0.871 74
		EHVI	100	1.3254e9 ± 1.8606e6	0.423 61 ± 0.225 54
		CEHVI	100	**1.3306e9** ± 2.3480e5	**0.028 47** ± 0.028 46
		NSGA2	200	1.3090e9 ± 4.2413e6	1.654 50 ± 0.514 12

Declarations

Conflict of interests. The authors declare that they have no conflict of interest.

References

[1] Močkus, J.: On bayesian methods for seeking the extremum. In: Marchuk, G.I. (ed.) Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974, pp. 400–404. Springer, Berlin, Heidelberg (1975)

[2] Mockus, J.B., Mockus, L.J.: Bayesian approach to global optimization and application to multiobjective and constrained problems. J. Optim. Theory Appl. **70**(1), 157–172 (1991)

[3] Garnett, R.: Bayesian Optimization. Cambridge University Press, Cambridge, United Kingdom (2022). in preparation

[4] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of Bayesian optimization (2016). https://doi.org/10.1109/JPROC.2015.2494218

[5] Jim, T.M., Faza, G.A., Palar, P.S., Shimoyama, K.: Bayesian optimization of a low-boom supersonic wing planform. AIAA journal **59**(11), 4514–4529 (2021)

[6] Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems **4**, 2951–2959 (2012) arXiv:1206.2944

[7] De Almeida, A.T., Ferreira, F.J.T.E., Duarte, A.Q.: Technical and economical considerations on super high-efficiency three-phase motors. IEEE Transactions on Industry Applications **50**(2), 1274–1285 (2014). https://doi.org/10.1109/TIA.2013.2272548

[8] Lei, G., Zhu, J., Guo, Y., Liu, C., Ma, B.: A Review of Design Optimization Methods for Electrical Machines. Energies **10**(12), 1962 (2017). https://doi.org/10.3390/en10121962. Accessed 2022-10-24

[9] Makni, Z., Besbes, M., Marchand, C.: Multiphysics design methodology of permanent-magnet synchronous motors. IEEE Transactions on Vehicular Technology **56**(4), 1524–1530 (2007). https://doi.org/10.1109/TVT.2007.896981

[10] Ibrahim, M.N., Sergeant, P.: Design and analysis of electric motor with integrated magnetic spring for cyclic loads. IEEE Transactions on Industrial Electronics, 1–9 (2022). https://doi.org/10.1109/TIE.2022.3210513

[11] Liu, X., Hu, C., Li, X., Gao, J., Huang, S.: An Online Data-Driven Multi-Objective Optimization of a Permanent Magnet Linear Synchronous Motor. IEEE Transactions on Magnetics **57**(7), 1–4 (2021). https://doi.org/10.1109/TMAG.2021.3059513. Accessed 2022-10-24

[12] Diao, K., Sun, X., Lei, G., Guo, Y., Zhu, J.: Multiobjective System Level Optimization Method for Switched Reluctance Motor Drive Systems Using Finite-Element Model. IEEE Transactions on Industrial Electronics **67**(12), 10055–10064 (2020). https://doi.org/10.1109/TIE.2019.2962483. Accessed 2022-10-24

[13] Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge, Massachusetts (2018). https://doi.org/10.7551/mitpress/3206.001.0001

[14] Seeger, M.: Gaussian Processes for Machine Learning, 52 (2004)

[15] Wiener, N.: The homogeneous chaos. American Journal of Mathematics **60**(4), 897–936 (1938). Accessed 2022-09-22

[16] Kaintura, A., Dhaene, T., Spina, D.: Review of Polynomial Chaos-Based Methods for Uncertainty Quantification in Modern Integrated Circuits. Electronics **7**(3), 30 (2018). https://doi.org/10.3390/electronics7030030. Accessed 2022-09-22

[17] Sun, G., Wang, S.: A review of the artificial neural network surrogate modeling in aerodynamic design. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering **233**(16), 5863–5872 (2019). https://doi.org/10.1177/0954410019864485

[18] Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M.M.A., Prabhat, P., Adams, R.P.: Scalable bayesian optimization using deep neural networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. ICML'15, pp. 2171–2180. JMLR.org, ??? (2015)

[19] Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011, 1–9 (2011)

[20] Ginsbourger, D., Le Riche, R., Carraro, L.: A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes (2008)

[21] Contal, E., Buffoni, D., Robicquet, A., Vayatis, N.: Parallel gaussian process optimization with upper confidence bound and pure exploration.

In: Blockeel, H., Kersting, K., Nijssen, S., Zelezný, F. (eds.) ECML/PKDD (1). Lecture Notes in Computer Science, vol. 8188, pp. 225–240. Springer, ??? (2013). https://doi.org/10.1007/978-3-642-40988-2_15. http://dblp.uni-trier.de/db/conf/pkdd/pkdd2013-1.html

[22] Kandasamy, K., Dasarathy, G., Oliva, J.B., Schneider, J., Póczos, B.: Gaussian process bandit optimisation with multi-fidelity evaluations. Advances in neural information processing systems **29** (2016)

[23] Gardner, J.R., Kusner, M.J., Jake, G.: Bayesian Optimization with Inequality Constraints, 9 (2014)

[24] Letham, B., Karrer, B., Ottoni, G., Bakshy, E.: Constrained Bayesian Optimization with Noisy Experiments. Bayesian Analysis **14**(2), 495–519 (2019). https://doi.org/10.1214/18-BA1110

[25] Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. Journal of Global Optimization **60**(3), 575–594 (2014). https://doi.org/10.1007/s10898-013-0118-2

[26] Emmerich, M.T.M., Deutz, A.H., Klinkenberg, J.W.: Hypervolume-based expected improvement: Monotonicity properties and exact computation. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 2147–2154. IEEE, New Orleans, LA, USA (2011). https://doi.org/10.1109/CEC.2011.5949880. http://ieeexplore.ieee.org/document/5949880/ Accessed 2022-09-26

[27] Wang, X., Jin, Y., Schmitt, S., Olhofer, M.: Recent Advances in Bayesian Optimization. arXiv (2022). https://doi.org/10.48550/ARXIV.2206.03301. https://arxiv.org/abs/2206.03301

[28] Allmendinger, R., Handl, J., Knowles, J.: Multiobjective optimization: When objectives exhibit non-uniform latencies. European Journal of Operational Research **243**(2), 497–513 (2015). https://doi.org/10.1016/j.ejor.2014.09.033. Accessed 2022-09-22

[29] Wang, X., Jin, Y., Schmitt, S., Olhofer, M.: Transfer Learning Based Co-surrogate Assisted Evolutionary Bi-objective Optimization for Objectives with Non-uniform Evaluation Times. arXiv. arXiv:2108.13339 [cs] (2021). http://arxiv.org/abs/2108.13339 Accessed 2022-09-23

[30] Wang, X., Jin, Y., Schmitt, S., Olhofer, M., Allmendinger, R.: Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times. Knowledge-Based Systems **227**, 107190 (2021). https://doi.org/10.1016/j.knosys.2021.107190. Accessed 2022-09-23

[31] Wang, X., Jin, Y., Schmitt, S., Olhofer, M.: Alleviating Search Bias in Bayesian Evolutionary Optimization with Many Heterogeneous Objectives. arXiv. arXiv:2208.12217 [cs] (2022). http://arxiv.org/abs/2208.12217 Accessed 2022-09-23

[32] Loka, N., Couckuyt, I., Garbuglia, F., Spina, D., Van Nieuwenhuyse, I., Dhaene, T.: Bi-objective Bayesian optimization of engineering problems with cheap and expensive cost functions. Engineering with Computers (2022). https://doi.org/10.1007/s00366-021-01573-7. Accessed 2022-09-22

[33] Yang, K., Emmerich, M., Deutz, A., Bäck, T.: Efficient computation of expected hypervolume improvement using box decomposition algorithms. Journal of Global Optimization **75**(1), 3–34 (2019). https://doi.org/10.1007/s10898-019-00798-7. Accessed 2022-08-29

[34] Diehl, O., Schönfeldt, M., Brouwer, E., Dirks, A., Rachut, K., Gassmann, J., Güth, K., Buckow, A., Gauß, R., Stauber, R., Gutfleisch, O.: Towards an Alloy Recycling of Nd–Fe–B Permanent Magnets in a Circular Economy. Journal of Sustainable Metallurgy **4**(2), 163–175 (2018). https://doi.org/10.1007/s40831-018-0171-7. Accessed 2023-02-07

[35] Elwert, T., Goldmann, D., Roemer, F., Schwarz, S.: Recycling of NdFeB Magnets from Electric Drive Motors of (Hybrid) Electric Vehicles. Journal of Sustainable Metallurgy **3**(1), 108–121 (2017). https://doi.org/10.1007/s40831-016-0085-1. Accessed 2023-02-07

[36] Luckeneder, S., Giljum, S., Schaffartzik, A., Maus, V., Tost, M.: Surge in global metal mining threatens vulnerable ecosystems. Global Environmental Change **69**, 102303 (2021). https://doi.org/10.1016/j.gloenvcha.2021.102303

[37] Meeker, D.: Finite element method magnetics

[38] Wang, J., Yuan, X., Atallah, K.: Design optimization of a surface-mounted permanent-magnet motor with concentrated windings for electric vehicle applications. IEEE Transactions on Vehicular Technology **62**(3), 1053–1064 (2013). https://doi.org/10.1109/TVT.2012.2227867

[39] Metwly, M.Y., Hemeida, A., Abdel-Khalik, A.S., Hamad, M.S., Ahmed, S.: Design and multi-objective optimization of a 12-slot/10-pole integrated obc using magnetic equivalent circuit approach. Machines **9**(12) (2021). https://doi.org/10.3390/machines9120329

[40] Edhah, S.O., Alsawalhi, J.Y., Al-Durra, A.A.: Multi-objective optimization design of fractional slot concentrated winding permanent magnet synchronous machines. IEEE Access **7**, 162874–162882 (2019)

[41] Silva, R.C.P., Rahman, T., Mohammadi, M.H., Lowther, D.A.: Multiple operating points based optimization: Application to fractional slot concentrated winding electric motors. IEEE Transactions on Industrial Electronics **65**(2), 1719–1727 (2018). https://doi.org/10.1109/TIE.2017.2756586

[42] Sarigiannidis, A.G., Beniakar, M.E., Kladas, A.G.: Fast adaptive evolutionary pm traction motor optimization based on electric vehicle drive cycle. IEEE Transactions on Vehicular Technology **66**(7), 5762–5774 (2016)

[43] Görtler, J., Kehlbeck, R., Deussen, O.: A visual exploration of gaussian processes. Distill (2019). https://doi.org/10.23915/distill.00017. https://distill.pub/2019/visual-exploration-gaussian-processes

[44] Fonseca, C.M., Paquete, L., Lopez-Ibanez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 1157–1163 (2006). https://doi.org/10.1109/CEC.2006.1688440

[45] Sheikh, H.M., Marcus, P.S.: Bayesian optimization for multi-objective mixed-variable problems. arXiv preprint arXiv:2201.12767 (2022)

[46] Daulton, S., Balandat, M., Bakshy, E.: Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. Advances in Neural Information Processing Systems **33**, 9851–9864 (2020)

[47] Daulton, S., Eriksson, D., Balandat, M., Bakshy, E.: Multi-objective bayesian optimization over high-dimensional search spaces. In: Uncertainty in Artificial Intelligence, pp. 507–517 (2022). PMLR

[48] Allmendinger, R., Handl, J., Knowles, J.: Multiobjective optimization: When objectives exhibit non-uniform latencies. European Journal of Operational Research **243**(2), 497–513 (2015). https://doi.org/10.1016/j.ejor.2014.09.033

[49] Allmendinger, R., Knowles, J.: Heterogeneous Objectives: State-of-the-Art and Future Research. arXiv (2021). https://doi.org/10.48550/ARXIV.2103.15546. https://arxiv.org/abs/2103.15546

[50] Buckingham, J.M., Gonzalez, S.R., Branke, J.: Bayesian Optimization of Multiple Objectives with Different Latencies (2023)

[51] Martínez-Frutos, J., Herrero-Pérez, D.: Kriging-based infill sampling criterion for constraint handling in multi-objective optimization. Journal of Global Optimization **64**(1), 97–115 (2016). https://doi.org/10.1007/

s10898-015-0370-8. Accessed 2023-01-05

[52] Viana, F.A.C., Venter, G., Balabanov, V.: An algorithm for fast optimal latin hypercube design of experiments. International Journal for Numerical Methods in Engineering **82**(2) (2010). https://doi.org/10.1002/nme.2750

[53] Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature — PPSN V, pp. 292–301. Springer, Berlin, Heidelberg (1998)

[54] Guerreiro, A.P., Fonseca, C.M., Paquete, L.: The Hypervolume Indicator: Problems and Algorithms. ACM Computing Surveys **54**(6), 1–42 (2021). https://doi.org/10.1145/3453474. arXiv:2005.00515 [cs]. Accessed 2022-09-27

[55] Emmerich, M., Yang, K., Deutz, A., Wang, H., Fonseca, C.: A Multicriteria Generalization of Bayesian Global Optimization, vol. 107. (2015). https://doi.org/10.1007/978-3-319-29975-4_12

[56] Muth, J.F., Thompson, G.L.: Industrial Scheduling. Prentice-Hall, Michigan (1963)

[57] Martí, R., Resende, M.G.C., Ribeiro, C.C.: Multi-start methods for combinatorial optimization. European Journal of Operational Research **226**(1), 1–8 (2013). https://doi.org/10.1016/j.ejor.2012.10.012

[58] Kraft, D.: A Software Package for Sequential Quadratic Programming. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, Germany (1988). https://books.google.be/books?id=4rKaGwAACAAJ

[59] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2

[60] Picheny, V., Berkeley, J., Moss, H.B., Stojic, H., Granta, U., Ober, S.W., Artemev, A., Ghani, K., Goodall, A., Paleyes, A., Vakili, S., Pascual-Diaz, S., Markou, S., Qing, J., Loka, N.R.B.S., Couckuyt, I.: Trieste: Efficiently Exploring The Depths of Black-box Functions with TensorFlow. arXiv

(2023). https://doi.org/10.48550/ARXIV.2302.08436. https://arxiv.org/abs/2302.08436

[61] Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer, London (2005). https://doi.org/10.1007/1-84628-137-7_6. Series Title: Advanced Information and Knowledge Processing

[62] Binh, T.T., Korn, U.: Mobes: A multiobjective evolution strategy for constrained optimization problems. In: IN PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS (MENDEL97, pp. 176–182 (1997)

[63] Chankong, V., Haimes, Y.Y.: Multiobjective Decision Making: Theory and Methodology. Dover Books on Engineering. Dover Publications, New York (2008). https://books.google.be/books?id=o371DAAAQBAJ

[64] Owen, A.B.: A randomized halton algorithm in r (2017)

[65] Durillo, J.J., Nebro, A.J., Luna, F., Alba, E.: On the effect of the steady-state selection scheme in multi-objective genetic algorithms. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **5467 LNCS**, 183–197 (2010). https://doi.org/10.1007/978-3-642-01020-0_18

[66] Blank, J., Deb, K.: pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)

[67] Ibrahim, M.N., Rezk, H., Al-Dhaifallah, M., Sergeant, P.: Modelling and design methodology of an improved performance photovoltaic pumping system employing ferrite magnet synchronous reluctance motors. Mathematics **8**(9) (2020). https://doi.org/10.3390/MATH8091429

[68] Ma, Q., El-Refaie, A., Lequesne, B.: Low-cost interior permanent magnet machine with multiple magnet types. IEEE Transactions on Industry Applications **56**(2), 1452–1463 (2020). https://doi.org/10.1109/TIA.2020.2966458

[69] Minasny, B., McBratney, A.B.: The Matérn function as a general model for soil variograms. Geoderma **128**(3-4 SPEC. ISS.), 192–207 (2005). https://doi.org/10.1016/j.geoderma.2005.04.003

[70] Norden, R.H.: A survey of maximum likelihood estimation. International Statistical Review / Revue Internationale de Statistique **40**(3), 329–354 (1972). Accessed 2022-07-25

[71] Emmerich, M., Yang, K., Deutz, A., Wang, H., Fonseca, C.M.: A multicriteria generalization of bayesian global optimization. In: Pardalos, P.M., Zhigljavsky, A., Žilinskas, J. (eds.) Advances in Stochastic and Deterministic Global Optimization. Springer Optimization and Its Applications, pp. 229–242. Springer, Switzerland (2016). https://doi.org/10.1007/978-3-319-29975-4

[72] Hupkens, I., Deutz, A., Yang, K., Emmerich, M.: Faster Exact Algorithms for Computing Expected Hypervolume Improvement. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) Evolutionary Multi-Criterion Optimization vol. 9019, pp. 65–79. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15892-1_5. Series Title: Lecture Notes in Computer Science