A MATLAB Toolbox for Surrogate-Assisted Multi-Objective Optimization: A Preliminary Study

Abdullah Al-Dujaili
School of Computer Science and Engineering
Nanyang Technological University
Singapore 639798
aldujail001@e.ntu.edu.sg

S. Suresh
School of Computer Science and Engineering
Nanyang Technological University
Singapore 639798
ssundaram@ntu.edu.sg

ABSTRACT

Surrogate modeling has been a powerful ingredient for several algorithms tailored towards computionally-expensive optimization problems. Concerned with solving black-box multi-objective problems given a finite number of function evaluations and inspired by the recent advances in multi-objective algorithms, this paper presents—based on the `MATSuMoTo` library for single-objective optimization—a surrogate-based optimization toolbox for multi-objective problems. Moreover, in attempt to highlight the strengths and weaknesses of the employed methods, we benchmark the presented toolbox within the Black-box Optimization Benchmarking framework (BBOB 2016).

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

Keywords

Benchmarking, Black-box optimization, Bi-objective optimization, Surrogate optimization, Response surface modeling

1. INTRODUCTION

Multi-objective Optimization Problems (MOPs)—in contrast to Single-objective Optimization Problems (SOPs)—involve a set of conflicting objectives that are to be optimized simultaneously. It has applications in various science, business, and engineering disciplines. Without loss of generality, an MOP with n decision variables and m objectives, has the form:

$$\begin{aligned} &\text{minimize} && \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ &\text{where} && \mathbf{x} = (x_1, \dots, x_n) \in \mathcal{X} \\ &&& \mathbf{y} = (y_1, \dots, y_m) \in \mathcal{Y} \end{aligned} \quad (1)$$

and where $\mathbf{x}$ is called the *decision vector (solution)*, $\mathbf{y}$ is called the *objective vector*, $\mathcal{X}$ is the *feasible decision space*, and $\mathcal{Y}$ is the corresponding *objective space*. In practice, it is common that derivatives of the objectives are neither symbolically nor numerically available [14], which makes such problems exceptionally tough to solve exactly, as the only source of information about the objective function is pointwise evaluation—hence they are commonly referred to as *black-box* problems. Furthermore, evaluating $\mathbf{f}$ is typically expensive requiring some computational resources (e.g., a computer code or a laboratory experiment). More specifically, we are asked to solve (1) using a finite budget of function evaluations.

Starting with the seminal paper of Jones *et al.* [17] on Efficient Global Optimization (EGO), optimization using surrogate models emerged as a powerful paradigm to approach computationally-expensive black-box SOPs as they require significantly fewer function evaluations (see [6] for more details). Likewise, surrogate methods have become increasingly popular for MOPs, though with different proposals about finding and handling a set of Pareto solutions rather than a single solution [1, 18, 26]. There are readily available software libraries for surrogate-assisted SOPs (e.g., [11,23]), which can be used by the community and that have been benchmarked on well-established test suites [7, 8, 24]. One the other hand, community efforts have been constantly growing towards *consolidating*—e.g., the recent SAMCO workshop[1]—and *benchmarking* surrogate-assisted algorithms for MOPs. To the best of our knowledge, most benchmarking efforts have been independently conducted on different sets of problems (see, e.g., [1, 26, 28]). This paper adds a brick to the ongoing effort by:

- Incorporating recent multi-objective techniques into a MATLAB toolbox for computationally-expensive black-box MOPs based on the `MATSuMoTo` (short for MATLAB Surrogate Model Toolbox) library [23].
- Providing an initial analysis of the presented toolbox's performance on the Bi-objective Black Box Optimization

[1] http://samco.gforge.inria.fr/doku.php

GECCO'16 Companion, July 20-24, 2016, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/2908961.2931703

Benchmarking (BBOB 2016 [29]) and compare it against the established surrogate-assisted multi-objective optimization algorithms, viz. SMS-EGO [26].

The BBOB benchmark [29] comes with a testbed of 55 scalable bi-objective problems addressing such real-world difficulties as ill-conditioning, multi-modality, and dimensionality. The rest of the paper is organized as follows. Section 2 provides a brief description of surrogate-assisted optimization and discusses the differentiating building blocks for MOPs with respect to SOPs. This motivates us—based on the MATSuMoTo library—to present, in Section 3, a MATLAB toolbox for surrogate-assisted multi-objective optimization, incorporating recent techniques for tackling computationally-expensive black-box MOPs. In Section 4, the numerical assessment of the presented methods is conducted on the BBOB platform: discussing the experimental setup, the procedure for evaluating the algorithms' performance, and elaborating the results. Section 5 summarizes the main points from this study.

2. SURROGATE-ASSISTED OPTIMIZATION

A typical surrogate-assisted optimization algorithm follows the six-step framework shown in Fig. 1. The problem can be treated as a sequential design, where the sample $\mathbf{x}^t \in \mathcal{X}$ at time t depends on the previous samples and their objectives' values $\{(\mathbf{x}^1, \mathbf{f}(\mathbf{x}^1)), \ldots, (\mathbf{x}^{t-1}, \mathbf{f}(\mathbf{x}^{t-1}))\}$. In essence, an initial design phase starts with sampling a few representative points in the decision space $\mathcal{X}$ (Step 1 in Fig. 1, e.g., Latin hypercube or random sampling [22]) and evaluating them on the expensive function $\mathbf{f}$ (Step 2). Then, an iterative procedure builds a *surrogate model* $\bar{\mathbf{f}}$ approximating $\mathbf{f}$ based on the already evaluated samples (Step 3), which is then used to nominate the next samples as $\bar{\mathbf{f}}$ is less expensive-to-evaluate than $\mathbf{f}$ (Step 4). The selected new samples are usually the optimizers of one criterion (or more) that extracts information from $\bar{\mathbf{f}}$ directly or through a derived measure. The criterion is known under as several names as "acquisition function", "infill criterion", "figure of merit", or "selection rule"; and its optimization is often done using available off-the-shelf optimization procedures (e.g., DIRECT [16]).

Although there have been recent propositions about surrogate models tailored towards MOPs (e.g., [20]), one can note that Step 4 is dominantly (see [1]) the main differentiating block for solving MOPs when compared to SOPs since the new selected points do not only need to approximate the optimizers of each objective, but also to capture the Pareto front akin to the MOP in hand. Furthermore, previous benchmarking studies concluded that the impact of the type of initial phase (Steps 1 and 2) is insignificant [7,8] and established a default setting for Step 3 [24,27] based on the cubic or Gaussian Radial Basis Functions (RBFs).

In looking for optimal solutions given a finite computation budget in a multi-objective setting, one seeks a balance between three search components, viz. *exploitation*: local search, *exploration*: global search, and *diversification*: well-spread Pareto solutions [9]. To this end, two distinctive approaches can be identified in selecting new evaluation points (Step 4 in Fig. 1):

- $A1$ Using the surrogate model indirectly to generate a set of candidate points: the selected points for evaluation are the optimizers of a measure derived from the surrogate model (e.g., [10, 25, 26, 28]).
- $A2$ Using the surrogate model directly to generate a set of candidate points: a subset of these points are then selected for evaluation based on a set of rules (e.g., [1, 21, 27]).

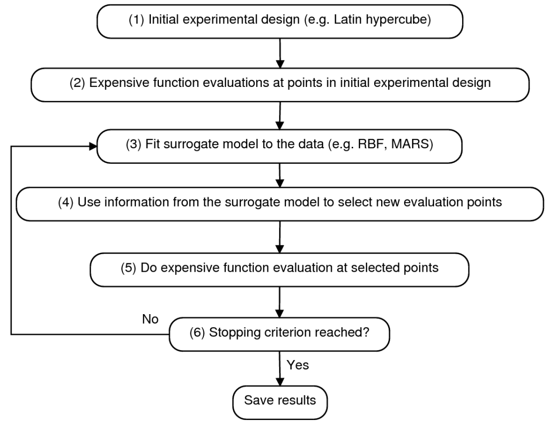

Figure 1: A generic framework for surrogate-assisted optimization. Adapted from [23].

While Approach $A1$ has been the focus of several optimization software packages (e.g., [5]), Approach $A2$ lends itself naturally to the framework of the MATSuMoTo library for SOPs [23]. In this paper, we incorporate a variant of Approach $A2$ into the MATSuMoTo library and assess its strength and weakness vs. a variant of Approach $A1$.

3. A MATLAB TOOLBOX FOR SURROGATE-ASSISTED MULTI-OBJECTIVE OPTIMIZATION

The MATLAB Surrogate Model Toolbox (MATSuMoTo) is a single-objective optimization software package for "*computationally expensive, black-box, global optimization problems that may have continuous, mixed-integer, or pure integer variables*" [23]. It comes with various configurations for the steps of Fig. 1, namely several choices for initial experimental design strategies, surrogate models, as well as strategies for selecting new evaluation points, which are in line with Approach $A2$. For this reason, we are motivated to adapt MATSuMoTo for MOPs by incorporating multi-objective $A2$ strategies as highlighted in Section 2.

In our survey for multi-objective $A2$ algorithms, we chose the recently proposed GOMORS (short for Gap Optimized Multi-Objective Optimization Using Response Surfaces) algorithm by Akhtar *et al.* [1] based on its reported result compared with other state-of-the-art algorithms. Besides its competitive performance, GOMORS provides a generic framework that naturally fits MATSuMoTo's architecture, where multiple rules for selecting new evaluation points (Step 4 in Fig. 1) can be employed promoting a balance among *exploration, exploitation, and diversification.* Examples of such rules are those based on the Euclidean distance among solutions in the decision and the objective spaces; they can be considered at once in an iteration or cycled through sequentially. It has been shown in [1] that using t rules to generate t candidate points at Step 4 is more effective than generating a single point by a single rule [1].

GOMORS typically employs an off-the-shelf evolutionary multi-objective solver on the built surrogate model $\bar{\mathbf{f}}$ to generate the set of candidate points. Here we use two available solvers for that purpose, viz. SMS-EMOA [4], and a randomized variant of MO-DIRECT [2]. Table 1 highlights the newly implemented features into MATSuMoTo supporting MOPs, namely the Gaussian RBF (motivated the numerical results of [27]), and the SurfPareto functionality which currently employs GOMORS to select new evaluations points.

Table 1: Possible feature choices for the individual steps of MATSuMoTo. Highlighted choices supports multi-objective optimization problems.

Algorithm Step	Choice Name	Description
(1) Initial design	CORNER	Corner points of the hypercube
	SLHD	Symmetric Latin hypercube
	lhd	Latin hypercube
(3) Surrogate model	RBFcub	Cubic RBF
	RBFgauss	Gaussian RBF
	RBFtps	Thin-plate spline RBF
	RBFlin	Linear RBF
	MARS	Multivariate adaptive regression spline
	POLYlin	Linear regression polynomial
	POLYquad	Quadratic regression polynomial
	POLYquadr	Reduced quadratic regression polynomial
	POLYcub	Cubic regression polynomial
	POLYcubr	Reduced cubic regression polynomial
	MIX_RcM	Mixture of RBFcub and MARS
	MIX_RcPc	Mixture of RBFcub and POLYcub
	MIX_RcPcr	Mixture of RBFcub and POLYcubr
	MIX_RcPq	Mixture of RBFcub and POLYquad
	MIX_RcPqr	Mixture of RBFcub and POLYquadr
	MIX_RcPcM	Mixture of RBFcub, POLYcub, and MARS
(4) Sampling strategy	CANDloc	Local candidate point search
	CANDglob	Global candidate point search
	SurfMin	Minimum point of surrogate model
	SurfPareto	Pareto front of surrogate model (currently employs GOMORS))

4. NUMERICAL VALIDATION

In this section, we investigate the efficacy of the proposed $A2$-based toolbox in solving computationally-expensive black-box problems and compare its performance with a representative algrotihm from Approach $A1$.

4.1 Compared Algorithms

To select a representative algorithm from Approach $A1$, we benchmarked several algorithms from GPareto R package [5], namely SMS-EGO [26], EHI-EGO [10], SUR-EGO [25], and EMI-EGO [28] on COCO v0.9. Running these algorithms on the COCO platform was extremely computationally-expensive. For instance, 5-D(imension) experiments with an evaluation budget $50 \cdot n$ took around five days for SUR-EGO, two days for EMI-EGO, and one day for SMS-EGO on Linux machine: Intel(R) Xeon(R) CPU E5-1650 0 @ 3.20GHz with 1 processor and 6 cores. On the other hand, EHI-EGO failed to finish.[2] We, therefore, chose SMS-EGO as the challenger. With regard to the proposed framework, we ran two instances employing SMS-EMOA [4] and a randomized variant of MO-DIRECT [2], respectively. We refer to them here by MAT-SMS and MAT-DIRECT, respectively.

4.2 Experimental Setup

We ran the three algorithms: SMS-EGO, MAT-SMS, and MAT-DIRECT on the COCO platform (v1.0) [29] for decision space dimensions $n \in \{2, 3, 5, 10, 20\}$.[3] As SMS-EGO (and the rest of tested $A1$ algorithms) is implemented in R [5], we had to write an unofficial binding for the COCO platfrom using the *R.matlab* R pacakge [3]. Consequently, the computational time needed for one function evaluation grew considerably (e.g., 5 days for SUR-EGO for 5-D problems). Such setup provided a simulated scenario of extremely expensive black-box optimization problems, with the goal of using as few function evaluations as possible. To this end and in the light of experimental settings from the literature [19], we recorded the performance of these algorithm on an evaluation budget of $75 \cdot n$ function evaluations.[4] This, however, made it difficult to interpret performance measures used by the COCO as its set targets are rather difficult to achieve given the *limited* computational budgets. Accordingly, we modified the default performance measures' target values to provide a better insight on the algorithms' performance. We used the default parameter settings for SMS-EGO and followed the guidelines of [1] in setting up MAT-SMS and MAT-DIRECT.

4.3 Performance Evaluation

The procedure for assessing the solution quality of an algorithm is based on recording its *runtime*: the number of function evaluations required by the algorithm for its solution to reach a specific (target) quality value [15]. The recorded runtimes are then expressed in terms of data profiles (or more accurately bootstrapped empirical cumulative distributions), which capture various aspects of the algorithms' convergence behavior. Furthermore, the average Running Time (aRT) is computed with respect to a given target value $\text{HV}_{target} = \text{HV}_{ref} + \Delta\text{HV}_{target}$, where *statistical significance* is tested with the rank-sum test for a given target HV_{target}.

4.4 Results

Results from experiments according to [12,13] on the benchmark functions given in [29] are shown in Figures 2 and 3 as data profiles; and Tables 2 and 3 reporting the aRT of the algorithms for each of the 55 problems. In general, both of MAT-SMS and MAT-DIRECT show a comparable performance outperforming the SMS-EGO algorithm from Approach $A1$ as shown cleary by the aRT values in Tables 2 and 3.

On the other hand, Figure 3 shows that SMS-EGO is better in its initial phase of search but its performance stagnates towards the end when compared with MAT-DIRECT and MAT-SMS whose performance gradually increases as a function of the number of function evaluations. This perhaps can be attributed to the restart strategy implemented within the toolbox framework, re-initiating the search whenever the accuracy of the built surrogate models in capturing the objectives' structure is decreased (e.g., ill-conditioned matrix for RBFs).

Gathering SMS-EGO's results took around one *week* distributed over 4 machines. On the other hand, the implemented toolbox took one day on a single machine for each of MAT-SMS and MAT-DIRECT. However, the comparison may be unfair due to the software wrapper that binds SMS-EGO with the COCO platform.

5. CONCLUSIONS

This paper has conducted a preliminary study on the extension of the MATLAB Surrogate Model Toolbox (MATSuMoTo)

[2]The algorithm exited with an error in executing the optim function

[3]SMS-EGO run on 20-D was not complete at the time of writing this paper.

[4]Nevertheless, we are in the process of running the algorithms with a $1000 \cdot n$-evaluation budget.

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f1					
MAT-DIR	**1**(0)	2.0(2)	144(295)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1.2**(0.5)	**91**(62)	∞ *375*	0/5
SMS-EGO	**1**(0)	1.4(1)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f2					
MAT-DIR	**1**(0)	14(17)	**157**(613)	∞ *375*	0/5
MAT-SMS	2.0(1)	**7.6**(10)	281(303)	∞ *375*	0/5
SMS-EGO	1.4(0.5)	37(44)	283(562)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f3					
MAT-DIR	1.4(0)	**3.2**(5)	107(188)	∞ *375*	0/5
MAT-SMS	**1**(0)	3.6(6)	**41**(10)	∞ *375*	0/5
SMS-EGO	4.4(8)	4.4(4)	103(14)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f4					
MAT-DIR	12(17)	**34**(4)	**320**(283)	∞ *375*	0/5
MAT-SMS	11(6)	39(14)	408(764)	∞ *375*	0/5
SMS-EGO	**2.6**(0.5)	97(188)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f5					
MAT-DIR	**1**(0)	**1**(0)	**7.8**(6)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	9.4(8)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	26(1)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f6					
MAT-DIR	2.6(2)	13(20)	**638**(1155)	∞ *375*	0/5
MAT-SMS	**1.4**(1)	10(9)	654(844)	∞ *375*	0/5
SMS-EGO	1.6(1)	**3.4**(2)	1620(2062)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f7					
MAT-DIR	2.4(1)	3.6(1)	**154**(292)	∞ *375*	0/5
MAT-SMS	2.6(2)	8.0(10)	181(102)	∞ *375*	0/5
SMS-EGO	**2.2**(2)	**3.4**(5)	1505(1594)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f8					
MAT-DIR	82(91)	**170**(281)	∞	∞ *375*	0/5
MAT-SMS	**36**(14)	200(240)	∞	∞ *375*	0/5
SMS-EGO	252(282)	1538(1594)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f9					
MAT-DIR	2.6(4)	**7.8**(2)	608(661)	∞ *375*	0/5
MAT-SMS	3.2(6)	27(66)	**454**(396)	∞ *375*	0/5
SMS-EGO	**1.4**(1)	95(281)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f10					
MAT-DIR	**1**(0)	2.4(2)	**507**(421)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1.2**(0.5)	740(469)	∞ *375*	0/5
SMS-EGO	**1**(0)	1.8(1)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f11					
MAT-DIR	**1**(0)	**1**(0)	**1**(0)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	1.4(0)	**1750**(1125)	1/5
SMS-EGO	**1**(0)	**1**(0)	1.2(0.2)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f12					
MAT-DIR	**1**(0)	**1**(0)	62(90)	∞ *375*	0/5
MAT-SMS	**1**(0)	1.2(0.5)	**57**(92)	∞ *375*	0/5
SMS-EGO	**1**(0)	1.2(0.5)	150(392)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f13					
MAT-DIR	**1**(0)	**1**(0)	**270**(478)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	570(1318)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	566(1033)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f14					
MAT-DIR	**1**(0)	**1**(0)	**2.4**(2)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	4.6(2)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	3.8(2)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f15					
MAT-DIR	13(15)	**25**(43)	**309**(800)	∞ *375*	0/5
MAT-SMS	13(28)	46(106)	373(375)	∞ *375*	0/5
SMS-EGO	**2.2**(2)	118(304)	1505(1594)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f16					
MAT-DIR	**1**(0)	**1**(0)	73(76)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	**59**(92)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	188(350)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f17					
MAT-DIR	95(282)	98(189)	**234**(261)	∞ *375*	0/5
MAT-SMS	**95**(188)	**95**(281)	284(588)	∞ *375*	0/5
SMS-EGO	95(188)	97(190)	684(875)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f18					
MAT-DIR	**1**(0)	**1**(0)	**4.0**(4)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	8.0(10)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	37(44)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f19					
MAT-DIR	1.4(0.5)	**1.8**(1)	**192**(61)	∞ *375*	0/5
MAT-SMS	**1.2**(0.5)	4.0(3)	382(356)	∞ *375*	0/5
SMS-EGO	**1.2**(0)	3.0(2)	1590(1875)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f20					
MAT-DIR	**1**(0)	**1**(0)	**95**(282)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	98(188)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	98(98)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f21					
MAT-DIR	2.2(3)	7.6(12)	**55**(50)	∞ *375*	0/5
MAT-SMS	1.4(0.5)	5.8(12)	290(295)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1.4**(1)	252(376)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f22					
MAT-DIR	**1**(0)	**1**(0)	2.0(2)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	1.8(1)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	**1.2**(0.5)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f23					
MAT-DIR	**1**(0)	2.4(2)	18(10)	∞ *375*	0/5
MAT-SMS	**1**(0)	1.6(2)	15(14)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	**12**(12)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f24					
MAT-DIR	2.2(2)	7.4(8)	147(227)	∞ *375*	0/5
MAT-SMS	3.8(6)	**6.4**(13)	**120**(401)	∞ *375*	0/5
SMS-EGO	**1.2**(0.5)	25(60)	121(114)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f25					
MAT-DIR	1.6(0.8)	**7.8**(8)	336(382)	∞ *375*	0/5
MAT-SMS	1.4(0.5)	8.0(15)	**179**(309)	∞ *375*	0/5
SMS-EGO	**1.2**(0.5)	19(22)	324(315)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f26					
MAT-DIR	**1**(0)	**1.2**(0.5)	**251**(562)	∞ *375*	0/5
MAT-SMS	**1**(0)	1.4(1)	252(656)	∞ *375*	0/5
SMS-EGO	**1**(0)	1.4(1)	252(750)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f27					
MAT-DIR	**1**(0)	**1**(0)	**11**(6)	∞ *375*	0/5
MAT-SMS	**1**(0)	1.2(0.5)	13(6)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	27(24)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f28					
MAT-DIR	**1**(0)	10(20)	308(469)	∞ *375*	0/5
MAT-SMS	**1**(0)	11(22)	**160**(394)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1.4**(1)	∞	∞ *375*	0/5

Table 2: Average running time (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2009 in dimension 5. The aRT and in braces, as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding best aRT in the first row. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $\mathbf{HV_{ref}} + 5 \times 10^{-1}$. The median number of conducted function evaluations is additionally given in italics, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of instances. A ↓ indicates the same tested against the best algorithm of BBOB-2009. Best results are printed in bold.

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f29					
MAT-DIR	10(20)	**20**(23)	**148**(131)	∞ *375*	0/5
MAT-SMS	6.8(7)	32(46)	175(146)	∞ *375*	0/5
SMS-EGO	**1.4**(1)	96(282)	1503(2625)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f30					
MAT-DIR	**20**(13)	**29**(5)	1874(844)	∞ *375*	0/5
MAT-SMS	40(46)	109(108)	∞	∞ *375*	0/5
SMS-EGO	253(281)	253(188)	**1701**(1312)	**1701**(1875)	1/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f31					
MAT-DIR	18(26)	78(26)	**469**(248)	∞ *375*	0/5
MAT-SMS	16(16)	**43**(44)	792(657)	∞ *375*	0/5
SMS-EGO	**1.8**(1)	170(94)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f32					
MAT-DIR	**27**(14)	**41**(12)	∞	∞ *375*	0/5
MAT-SMS	84(59)	196(238)	∞	∞ *375*	0/5
SMS-EGO	370(557)	1771(1406)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f33					
MAT-DIR	3.2(4)	15(4)	1561(2156)	∞ *375*	0/5
MAT-SMS	2.4(2)	17(20)	1541(2250)	∞ *375*	0/5
SMS-EGO	**1.4**(1)	**1.8**(1)	**1503**(1688)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f34					
MAT-DIR	21(12)	**36**(13)	∞	∞ *375*	0/5
MAT-SMS	19(14)	42(20)	**1740**(2138)	∞ *375*	0/5
SMS-EGO	**2.6**(1)	253(656)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f35					
MAT-DIR	**1**(0)	**1**(0)	**1**(0)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	**1**(0)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	**1**(0)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f36					
MAT-DIR	**1**(0)	**1**(0)	125(108)	∞ *375*	0/5
MAT-SMS	**1**(0)	1.2(0)	**119**(188)	∞ *375*	0/5
SMS-EGO	**1**(0)	1.4(0.5)	255(562)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f37					
MAT-DIR	1.4(0)	3.0(3)	**17**(25)	∞ *375*	0/5
MAT-SMS	2.4(2)	4.4(3)	22(17)	∞ *375*	0/5
SMS-EGO	**1.2**(0.2)	**1.4**(0.5)	104(379)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f38					
MAT-DIR	1.8(1)	10(22)	252(268)	∞ *375*	0/5
MAT-SMS	2.0(2)	9.0(16)	**173**(416)	∞ *375*	0/5
SMS-EGO	**1.4**(1)	**7.6**(1)	1720(2719)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f39					
MAT-DIR	**1**(0)	**1**(0)	**39**(38)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	**39**(37)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	254(282)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f40					
MAT-DIR	**1**(0)	**1**(0)	39(8)	∞ *375*	0/5
MAT-SMS	**1**(0)	**1**(0)	**37**(26)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1**(0)	579(482)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f41					
MAT-DIR	22(46)	157(122)	∞	∞ *375*	0/5
MAT-SMS	5.4(10)	**34**(30)	∞	∞ *375*	0/5
SMS-EGO	**2.4**(2)	252(656)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f42					
MAT-DIR	**2.8**(2)	**32**(57)	**1566**(750)	∞ *375*	0/5
MAT-SMS	5.0(4)	118(210)	1699(2156)	∞ *375*	0/5
SMS-EGO	11(8)	114(108)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f43					
MAT-DIR	**19**(30)	**148**(376)	∞	∞ *375*	0/5
MAT-SMS	100(192)	291(471)	∞	∞ *375*	0/5
SMS-EGO	102(188)	579(1125)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f44					
MAT-DIR	**1**(0)	35(32)	**1546**(1500)	∞ *375*	0/5
MAT-SMS	**1**(0)	23(20)	1867(2062)	∞ *375*	0/5
SMS-EGO	**1**(0)	**3.8**(2)	1729(2156)	**1729**(1594)	1/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f45					
MAT-DIR	7.4(14)	20(20)	813(946)	∞ *375*	0/5
MAT-SMS	1.8(0.8)	**19**(20)	**752**(893)	∞ *375*	0/5
SMS-EGO	**1.4**(0.5)	95(1)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f46					
MAT-DIR	19(28)	**50**(35)	1686(1406)	∞ *375*	0/5
MAT-SMS	**3.0**(2)	106(291)	∞	∞ *370*	0/5
SMS-EGO	4.6(4)	119(112)	**1505**(2156)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f47					
MAT-DIR	14(28)	**101**(186)	**1867**(1312)	∞ *375*	0/5
MAT-SMS	10(19)	132(128)	∞	∞ *375*	0/5
SMS-EGO	**3.0**(2)	104(99)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f48					
MAT-DIR	1.4(1)	46(69)	∞	∞ *375*	0/5
MAT-SMS	**1.2**(0.5)	31(29)	∞	∞ *375*	0/5
SMS-EGO	1.8(2)	**4.2**(2)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f49					
MAT-DIR	4.2(3)	10(2)	**1552**(1219)	∞ *375*	0/5
MAT-SMS	2.0(1)	9.4(14)	1846(2625)	∞ *375*	0/5
SMS-EGO	**1.8**(2)	**2.2**(1)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f50					
MAT-DIR	118(204)	**355**(542)	∞	∞ *375*	0/5
MAT-SMS	17(16)	367(386)	∞	∞ *375*	0/5
SMS-EGO	**6.2**(7)	1503(2344)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f51					
MAT-DIR	2.2(0.5)	13(11)	∞	∞ *375*	0/5
MAT-SMS	1.6(0.8)	13(16)	**1643**(1029)	∞ *375*	0/5
SMS-EGO	**1**(0)	**3.4**(2)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f52					
MAT-DIR	5.6(2)	**60**(58)	∞	∞ *375*	0/5
MAT-SMS	7.6(4)	79(64)	∞	∞ *375*	0/5
SMS-EGO	**1.8**(0.5)	578(1031)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f53					
MAT-DIR	2.4(4)	**5.0**(10)	**295**(375)	∞ *375*	0/5
MAT-SMS	2.6(4)	8.0(9)	317(128)	∞ *375*	0/5
SMS-EGO	**1.4**(1)	95(188)	1505(2062)	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f54					
MAT-DIR	**1**(0)	**1.8**(1)	**1773**(1031)	∞ *375*	0/5
MAT-SMS	**1**(0)	3.6(3)	∞	∞ *375*	0/5
SMS-EGO	**1**(0)	**1.8**(1)	∞	∞ *375*	0/5

Δf_{opt}	1e2	1e1	1e0	5e-1	#succ
f55					
MAT-DIR	**1**(0)	4.2(8)	813(844)	∞ *371*	0/5
MAT-SMS	**1**(0)	2.8(4)	**674**(851)	∞ *375*	0/5
SMS-EGO	**1**(0)	**1.4**(0.5)	∞	∞ *375*	0/5

Table 3: Average running time (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2009 in dimension 5. The aRT and in braces, as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding best aRT in the first row. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $\mathbf{HV_{ref}} + 5\times10^{-1}$. The median number of conducted function evaluations is additionally given in italics, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of instances. A ↓ indicates the same tested against the best algorithm of BBOB-2009. Best results are printed in bold.

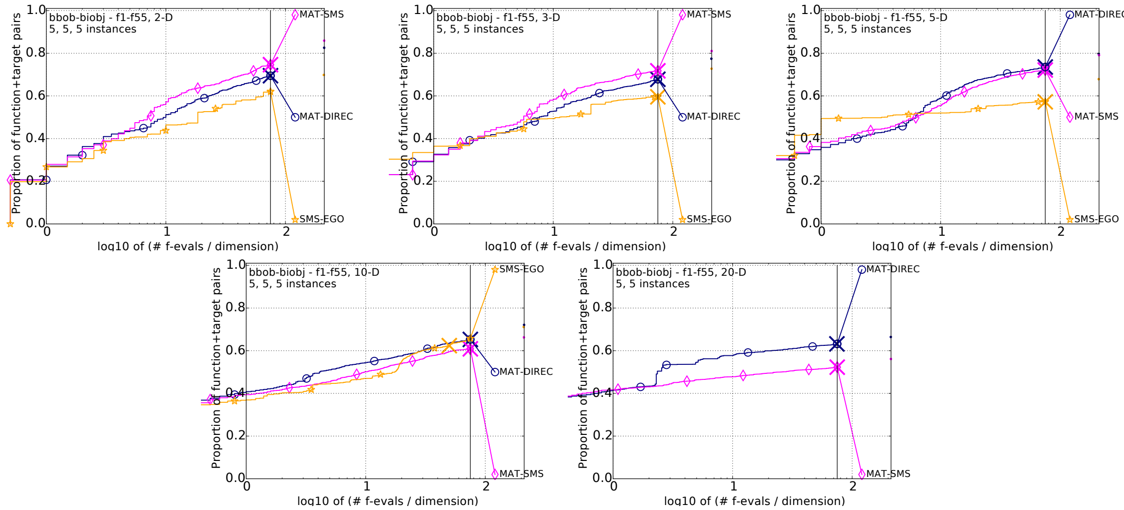

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 121 targets with target precision in $\{0, 10^{-0.19}, 10^{-0.18}, \ldots, 10^{0.98}, 10^{0.99}, 10^{1}\}$ over all the problems in $n \in \{2, 3, 5, 10, 20\}$.

to solve computationally-expensive black-box multi-objective optimization problems. The efficacy of the proposed framework has been validated on the COCO platform with 55 bi-objective problems and compared with other established surrogate-assisted multi-objective optimization algorithms.

6. ACKNOWLEDGMENTS

The authors would like to thank National Research Foundation (NRF), Singapore, project #NRF-ENIC-SERTD-SMES-NTUJTCI3C-2016, SMES, for providing financial support to conduct this study.

7. REFERENCES

[1] T. Akhtar and C. A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization*, 64(1):17–32, 2015.

[2] A. Al-Dujaili and S. Suresh. Dividing rectangles attack multi-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC), 2016*. IEEE, 2016.

[3] H. Bengtsson. *R.matlab: Read and Write MAT Files and Call MATLAB from Within R*. R package version 3.5.0., 2016.

[4] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[5] M. Binois and V. Picheny. *GPareto: Gaussian Processes for Pareto Front Estimation and Optimization*, 2016. R package version 1.0.2.

[6] A. J. Booker, J. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13, 1999.

[7] D. Brockhoff. Comparison of the MATSuMoTo library for expensive optimization on the noiseless black-box optimization benchmarking testbed. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 2026–2033, May 2015.

[8] D. Brockhoff, B. Bischl, and T. Wagner. The impact of initial designs on the performance of MATSuMoTo on the noiseless bbob-2015 testbed: A preliminary study. In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, pages 1159–1166. ACM, 2015.

[9] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

[10] M. Emmerich, A. H. Deutz, and J. W. Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2147–2154. IEEE, 2011.

[11] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq. A surrogate modeling and adaptive sampling toolbox for computer based design. *The Journal of Machine Learning Research*, 11:2051–2055, 2010.

[12] N. Hansen, A. Auger, O. Mersmann, T. Tusar, and D. Brockhoff. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *arXiv preprint arXiv:1603.08785*, 2016.

[13] N. Hansen, T. Tusar, A. Auger, D. Brockhoff, and O. Mersmann. COCO: Experimental Procedure. http://numbbo.github.io/coco-doc/experimental-setup/, 2016.

[14] A. L. Hoffmann, A. Y. Siem, D. den Hertog, J. H. Kaanders, and H. Huizenga. Derivative-free generation and interpolation of convex pareto optimal IMRT plans. *Physics in medicine and biology*, 51(24):6349, 2006.

[15] H. H. Hoos and T. Stützle. Evaluating las vegas algorithms: pitfalls and remedies. In *Proceedings of the Fourteenth conference on Uncertainty in artificial*

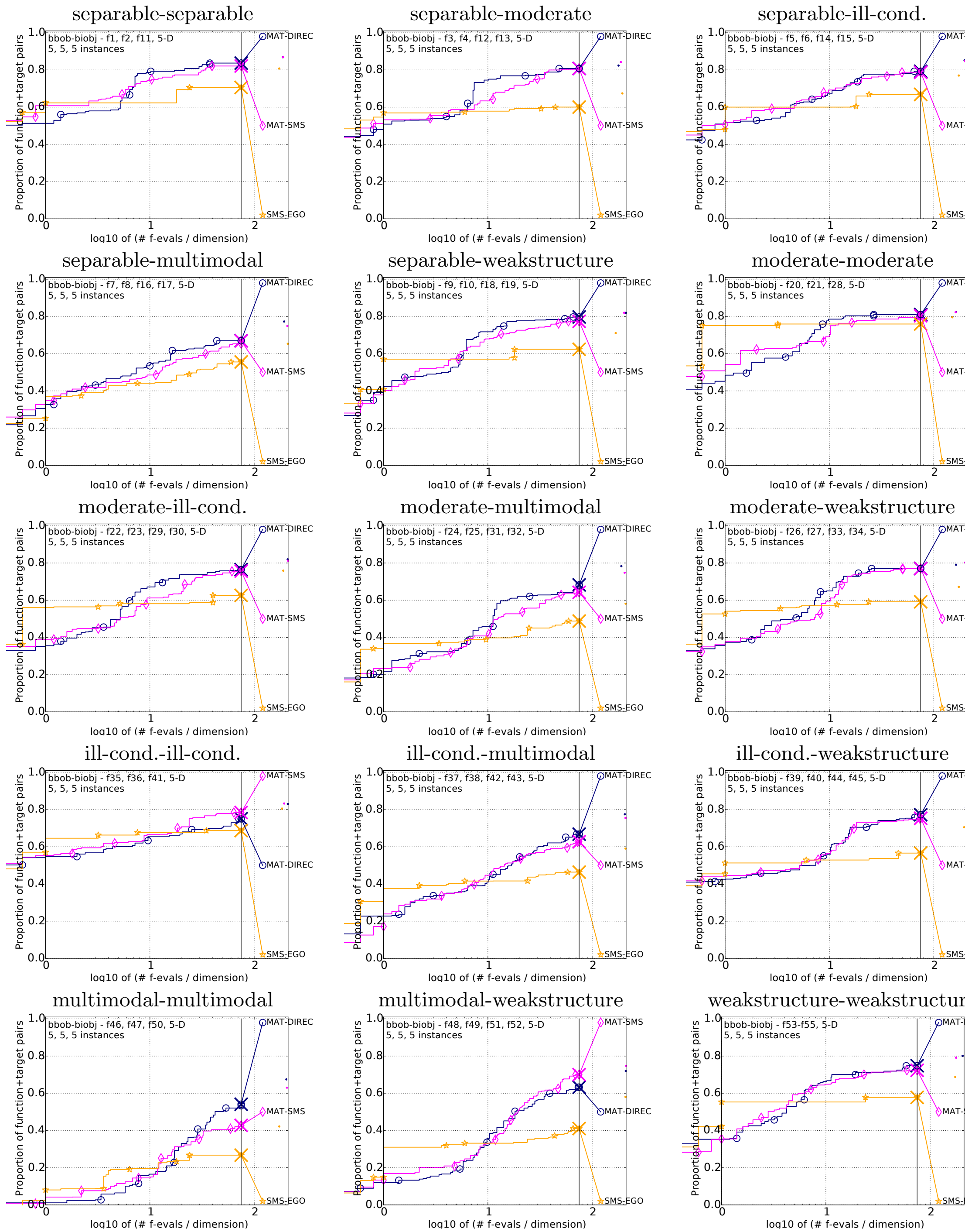

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 121 targets with target precision in $\{0, 10^{-0.19}, 10^{-0.18}, \ldots, 10^{0.98}, 10^{0.99}, 10^{1}\}$ for all function subgroups in 5-D.

intelligence, pages 238–245. Morgan Kaufmann Publishers Inc., 1998.

[16] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[17] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[18] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, Feb 2006.

[19] J. Knowles and E. J. Hughes. Multiobjective optimization on a budget of 250 evaluations. In *Evolutionary Multi-Criterion Optimization*, pages 176–190. Springer, 2005.

[20] I. Loshchilov, M. Schoenauer, and M. Sebag. A mono surrogate for multiobjective optimization. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 471–478. ACM, 2010.

[21] S. Z. MartÃŇnez and C. A. C. Coello. Combining surrogate models and local search for dealing with expensive multi-objective optimization problems. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2572–2579, June 2013.

[22] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[23] J. Mueller. MATSuMoTo: The matlab surrogate model toolbox for computationally expensive black-box global optimization problems. *arXiv preprint arXiv:1404.4261*, 2014.

[24] J. Müller and C. A. Shoemaker. Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123–144, 2014.

[25] V. Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.

[26] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *Parallel Problem Solving from Nature–PPSN X*, pages 784–794. Springer, 2008.

[27] L. V. Santana-Quintero, V. A. Serrano-Hernandez, C. A. C. Coello, A. G. Hernandez-Diaz, and J. Molina. Use of radial basis functions and rough sets for evolutionary multi-objective optimization. In *Computational Intelligence in Multicriteria Decision Making, IEEE Symposium on*, pages 107–114, April 2007.

[28] J. D. Svenson and T. J. Santner. Multiobjective optimization of expensive black-box functions via expected maximin improvement. *The Ohio State University, Columbus, Ohio*, 2010.

[29] T. Tusar, D. Brockhoff, N. Hansen, and A. Auger. COCO: The Bi-objective Black Box Optimization Benchmarking (bbob-biobj) Test Suite. ArXiv e-prints, arXiv:1604.00359, Apr. 2016.