Model-Based Multi-Objective Optimization: Taxonomy, Multi-Point Proposal, Toolbox and Benchmark

Daniel Horn[1]*, Tobias Wagner[2], Dirk Biermann[2], Claus Weihs[1], and Bernd Bischl[1]

[1] Technische Universität Dortmund, Chair of Computational Statistics, Vogelpothsweg 87, 44227 Dortmund
{dhorn,weihs,bischl}@statistik.uni-dortmund.de
[2] Technische Universität Dortmund, Institute of Machining Technology (ISF), Baroper Str. 303, 44227 Dortmund
{wagner,biermann}@isf.de

Abstract. Within the last 10 years, many model-based multi-objective optimization algorithms have been proposed. In this paper, a taxonomy of these algorithms is derived. It is shown which contributions were made to which phase of the MBMO process. A special attention is given to the proposal of a set of points for parallel evaluation within a batch. Proposals for four different MBMO algorithms are presented and compared to their sequential variants within a comprehensive benchmark. In particular for the classic ParEGO algorithm, significant improvements are obtained. The implementations of all algorithm variants are organized according to the taxonomy and are shared in the open-source R package mlrMBO.

Keywords: Expected Improvement, Hypervolume, Kriging, Performance Indicator, Surrogate Model

1 Introduction

In recent years, the use of surrogate models for partly replacing the actual objective function allowed multi-objective optimization techniques to be applied to real-world problems in an efficient way [16]. The resulting combinations of surrogate models and optimization algorithms are denoted as model-based multi-objective optimization (MBMO) algorithms in the following. In the early algorithms, surrogate models have been fitted, and have then been used for the

* We acknowledge partial support by the Mercator Research Center Ruhr under grant Pr-2013-0015 *Support-Vektor-Maschinen für extrem große Datenmengen*, by the German Research Foundation (DFG) within the Collaborative Research Center SFB 823 *Statistical modelling of nonlinear dynamic processes*, project C2, and within the Collaborative Research Center SFB 708 *3D-Surface Engineering*, project C4. In addition, the authors acknowledge support by the French national research agency (ANR) within the Modèles Numérique project NumBBO – Analysis, Improvement and Evaluation of Numerical Blackbox Optimizers.

optimization in replacement of the actual objective functions. No sequential update has been performed. If a validation is performed at all, only the finally selected solution has been evaluated on the actual problem.

Since 2005, sequential approaches – using the surrogate to decide on new points to evaluate and update the model in an iterative fashion – have been proposed. Most of these approaches are based on ideas of the popular Efficient Global Optimization (EGO) procedure [13]. Early work in the multi-objective scenario has either scalarized the objectives [15] to allow EGO to be directly used or has optimized EGO's figure of merit for different models in parallel using MOEA [11,19]. Later, also set-based improvement criteria, specifically designed for multi-objective optimization, have been defined [1,9,14,18,23]. Until now, the algorithms as a whole were considered as a contribution to the field of MBMO. In order to better distinguish the actual contributions, a first taxonomy of existing MBMO approaches is introduced in this paper.

Due to the enormous growth of parallel computing power and the advantages of performing real experiments in batches, allowing more than one point to be proposed per iteration (batch processing) is of great interest. Right now, only one multi-objective approach exists [23] (see [3] for a comparison of methods and a new approach in the single-objective case). As a consequence, possibilities to integrate batch proposals into existing MBMO algorithms are proposed in the paper. In particular for set-based improvement criteria in MBMO, this is done for the first time, to the best of our knowledge.

The taxonomy is introduced in section 2. In section 3, it is shown how the existing algorithms can be classified using the concepts of the taxonomy. The ideas for allowing a batch proposal within specific algorithm classes are proposed in section 4. All covered algorithms are integrated into the R toolbox mlrMBO for model-based optimization (MBO), whose software design closely reflects the presented taxonomy. The toolbox is briefly presented in section 5. The MBMO algorithms are compared on a comprehensive benchmark, which is described and evaluated in section 6. The paper is concluded by a summary of the results and an outlook on possible further improvements.

2 Taxonomy

The taxonomy of the MBMO approaches is based on the standard procedure of a sequential MBO algorithm, whose phases are shown on the left of Fig. 1. First, an initial design is evaluated on the actual, expensive objective function in order to train the surrogate model. In principle, all available design-of-experiment (DOE) techniques can be used. Due to its connection to the established Kriging models [13], however, Latin Hypercube Sampling (LHS) is applied in almost all existent MBMO approaches, and hence explicitly mentioned as an option.

For model fitting, two approaches are established. In the straightforward variant, an individual surrogate model is built for each objective function. In order to allow established single-objective model-based optimization criteria [12], such as the expected improvement (EI), the probability of improvement (PI) or

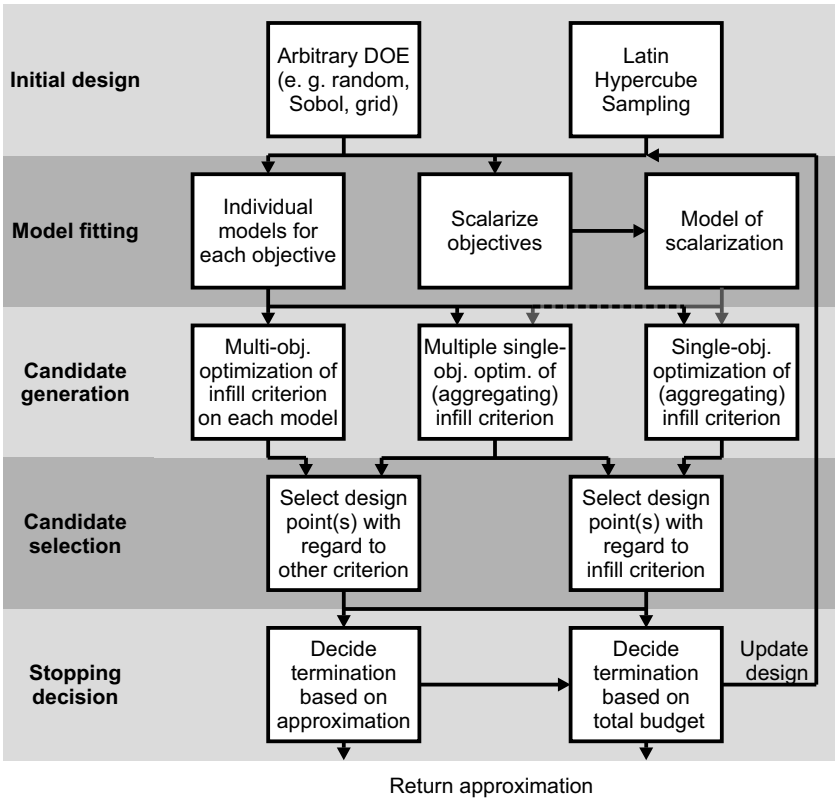

Fig. 1. Phases and tasks within a generalized MBMO algorithm

the lower confidence bound (LCB), to be directly used, the objectives can be scalarized before the surrogate model is fitted. In this case, the multi-objective problem is effectively reduced to a single-objective one. To still obtain an approximation of the complete Pareto frontier, the parameterization or type of the scalarization function can be varied over the iterations of the MBMO algorithm.

The candidate generation represents the step where most of the contributions have been made. In case of a single model, which predicts the value of a scalarization of the objectives, established criteria for generating candidates can be used [3, 10, 12]. If individual models for each objective are available, three different strategies can be pursued. Due to the current focus on single-point proposals and set-based multi-objective optimization, mainly criteria for an internal single-objective optimization of an aggregating infill criterion on the model [21] have been proposed within recent work. Also specific algorithms for performing the internal optimization have been designed [20]. A single optimum solution is found, which is then evaluated in order to update the training set for the model.

If a batch of solutions is desired, two alternative options can be used. In the first one, the internal optimization is performed by an MOEA, which op-

erates on single-objective infill criteria for each model. The final Pareto front approximation then provides the candidates for the batch processing. In the second variant, different single-objective subproblems (e.g. scalarizations) are optimized in parallel based on an established infill criterion. By collecting the respective optimum solutions, a batch of candidates is compiled. This approach has only been performed within a single algorithm [23] until now.

In the last step, the candidate set is reduced to the desired size. It is thus only required in case of a multi-point proposal in the candidate generation step. As the outcome of the MOEA can be filtered to obtain a mutually nondominated set, another, aggregating infill criterion has to be chosen for the selection if the number of solutions exceeds the desired one. Compared to the direct optimization of this criterion, the MOEA allows multiple points to be found, improves exploration of the search space and prevents effects of oversearching. On the other hand, it may result in suboptimal solutions with regard to the final selection criterion. If more subproblems than the desired batch size have been internally optimized, potentially two selection approaches can be used. The former uses a similar, global variant of the internal criteria within in the subproblems, whereas the latter decides based on a completely different, e.g., space-filling, explorative criterion. Both approaches improve exploration while also retaining the optimality, at least with regard to the defined subproblems.

After each iteration, it is checked whether the optimization process can be terminated. This decision is usually based on a budget of evaluations fixed beforehand. Recently, however, a new method to estimate the uncertainty of the current Pareto front approximation has been proposed [2]. In case the desired approximation quality is obtained before the total budget is spent, the remaining evaluations can be skipped to save expensive resources.

3 Considered MBMO algorithms

In this subsection, the algorithms considered in the following benchmark study are described as instantiations of the taxonomy. We only omitted algorithms applying complex and tedious indicator-based improvement criteria (cf. paragraph on Direct Indicator-Based MBMO), as recent studies have shown that conceptually similar (with regard to the taxonomy), but computationally cheaper variants, provide a comparable or even better performance [20]. By these means, the generality of the taxonomy is demonstrated. A summary of all approaches and their classification is provided in Table 1. All algorithms optimize a function $\boldsymbol{f} : \mathbf{R}^d \to \mathbf{R}^m$, where d is the decision space dimension and m the number of objective functions. For specific evaluation, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}$ holds.

Scalarization-Based MBMO Two scalarization-based MBMO algorithms using the augmented Tchebycheff norm

$$u(x) = -\max\left[\boldsymbol{w}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{i})\right] + \rho \boldsymbol{w}^T(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{i}) \tag{1}$$

Table 1. Summary of the approaches from literature considered in the benchmark

Algorithm	**Initial design**	**Model fitting**	**Candidate Generation**	**Candidate selection**	**Stopping decision**
ParEGO [15]	LHS	Model of scalarization	single-objective optimization of EI	One point, same criterion	Total budget
MOEA/D-EGO [23]	LHS	Models for each objective	Multiple single-objective optimizations of scalarizations	Multi point, same criterion (on clusters of subproblems)	Total budget
Multi-EGO [11]	LHS	Models for each objective	Multi-objective optimization of individual EI	Multi point, space-filling selection	Total budget
MOEA using Surrogates [19]	Other DOE (Sobol)	Models for each objective	Multi-objective optimization of model prediction	Multi point, space-filling selection	Total budget
MSPOT [22]	LHS	Models for each objective	Multi-objective optimization of model prediction	One point, hypervolume contribution	Total budget
SMS-EGO [18]	LHS	Models for each objective	Single-objective optimization of the hypervolume contribution	One point, same criterion	Total budget
ε-EGO [20]	LHS	Models for each objective	Single-objective optimization of the additive ε-indicator	One point, same criterion	Total budget

with ideal point $\boldsymbol{i}$ and weight vector $\boldsymbol{w}$ ($\sum_{j=1}^{m} w_j = 1$) do exist, which differ in model fitting, candidate generation, and candidate selection. ParEGO [15] randomly chooses $\boldsymbol{w}$ from a uniformly distributed set in each iteration. The surrogate model is fitted to the respective scalarization, and the EI is optimized on this model. Only the optimum solution is evaluated on the actual problem. In contrast, MOEA/D-EGO [23] fits models for each objective. In the internal optimization, the EI for all weight vectors is maximized in parallel, and finally the solutions obtaining the highest EI within N predefined weight vector clusters are evaluated. As a consequence, the distribution of chosen solutions with respect to the corresponding $\boldsymbol{w}$ can suffer a bias towards balanced components [7].

Pareto-Based MBMO The algorithms summarized under the term Pareto-based MBMO are using a multi-objective optimization of infill criteria on each objective in order to obtain a candidate set for evaluation. In Multi-EGO [11], the EI is used, whereas MSPOT [22] or Voutchkov's and Keane's surrogate-based MOEA [19] directly optimize the model predictions $\hat{\boldsymbol{y}}(\boldsymbol{x})$. The final selection from the Pareto front approximation is either distance- (Multi-EGO, Surrogate-based MOEA) or indicator-based (MSPOT).

Direct Indicator-Based MBMO In indicator-based MBMO algorithms, the contribution of an additional point to the indicator value of the current Pareto front approximation $\mathcal{Y}^*_{\text{approx}}$ is formulated as a single-objective criterion for the internal optimization. In the literature, two approaches can be distinguished. The first ones directly evaluate $\hat{\boldsymbol{y}}(\boldsymbol{x})$ or simple combinations with the associated uncertainty $\hat{\boldsymbol{s}}(\boldsymbol{x})$, such as the LCB $l(\boldsymbol{x}) = \hat{\boldsymbol{y}}(\boldsymbol{x}) + \lambda\hat{\boldsymbol{s}}(\boldsymbol{x})$. The algorithms are denoted as direct indicator-based (DIB) approaches. Examples are the SMS-EGO [18] for the hypervolume and the ε-EGO [20] for the additive ε-indicator. Whereas the former is enhanced by a check for nondominance based on additive ε-dominance ($\preceq_\varepsilon$), i.e., nondominance by an additional gap of ε, and a respective penalty $\Psi(\boldsymbol{x}) = \max_{\{\boldsymbol{y}^{(i)} \in \mathcal{Y}^*_{\text{approx}} | \boldsymbol{y}^{(i)} \preceq_\varepsilon l(\boldsymbol{x})\}} -1 + \prod_{j=1}^{m} \Big(1 + \max(l_j(\boldsymbol{x}) - y_j^{(i)}, 0)\Big)$, the latter only uses the respective indicator.

In addition, also more complex indicator-based infill criteria have been proposed. Their criteria analytically compute the expected improvement of the respective indicator by tediously integrating over the objective space [1,9]. Despite improvements regarding their complexity [8], these indicators are hard to implement. They are thus excluded from this benchmark study.

4 Batch Proposal for Parallel Evaluation

The structure implied by the taxonomy allows the realization of single phases to be easily replaced. This was used to propose N points for a batch evaluation within different MBMO algorithms originally designed for a single-point setup. To accomplish this, the candidate generation and selection steps of these algorithms were modified.

4.1 ParEGO

ParEGO was enhanced to a multi-point proposal by increasing the number of weight vectors randomly drawn in each iteration. If N points are desired, cN ($c > 1$) weight vectors are selected. Then, the pairwise distance between all weight vectors is calculated, and one vector of the pair resulting in the minimum distance is eliminated. This procedure is repeated until the set is reduced to the desired size. This greedy reduction of the larger set ensures that the selected weights cover the weight space in an almost uniform way.

In the following, the scalarizations implied by each weight vector are computed and individual models for each scalarization are fitted and optimized with respect to a single-objective infill criterion. The respective optima of each model build the batch to be evaluated. As the fitting and optimization of each model are mutually independent, they can be computed in parallel.

4.2 Pareto-Based MBMO

For the Pareto-based MBMO algorithms, the candidate generation by means of an internal multi-objective optimization already produces enough candidates

for the batch evaluation. In the algorithms relying on a distance-based selection [11, 19], a multi-point proposal is already realized. Hence, particularly the indicator-based candidate selection of MSPOT was enhanced to a multi-point proposal. To accomplish this, a greedy selection was used. Until the number of desired candidates for the batch evaluation is reached, the point of the candidate set having the highest contribution to the indicator is selected, added to the Pareto front approximation, and the contributions of the remaining points is updated. Consequently, the advantage of the multi-objective candidate generation to produce a set instead of single points is not only used for improving the exploration of the decision space, but also for obtaining a well-spread batch of solutions.

4.3 Direct Indicator-Based MBMO

For integrating a multi-point proposal within SMS- and ε-EGO, the concept of simulated evaluations was used. The optimization of the respective infill criterion is performed in its standard way, but the optimum solution is not directly evaluated on the actual, expensive problem. Instead, the LCB $l(\boldsymbol{x}^*)$ of the optimum solution $\boldsymbol{x}^*$ is added to the current Pareto front approximation without refitting the model. Based on the updated approximation, the criterion is optimized again, and the procedure is repeated until N points for a batch evaluation have been found. As the contribution to the indicator in the vicinity of the simulated point vanishes, particularly due to the optimistic bias implied by the LCB, it is likely that the following optimization will focus on different areas of the objective space. Hence, a batch of solutions distributed over the Pareto front is expected.

5 The mlrMBO R Software Package

The mlrMBO package [4] is based on the mlr package for machine learning in R [6]. It is designed as an encompassing toolbox for general MBO techniques, including single- and multi-objective, as well as single- and multi-point methods. Not only Kriging can be used as a surrogate model, but every regression method integrated into mlr. In the single-objective case, the package allows the optimization of mixed decision spaces, including integer, categorical and dependent parameters[3]. Extensive logging into a well structured archive enables the post-hoc inspection of runs. The archive contains the Pareto front and set, as well as all evaluations made in optimization process. By these means, visualizations of the runs are possible (at least for bi-objective problems). This is useful for a deeper understanding of algorithmic aspects in order to derive potential improvements. As real-world runs on, e.g., complicated simulators, often introduce technical problems, the package contains various error-handling mechanisms.

The setup of an MBMO algorithm by means of the toolbox is done by special control objects which closely follow the structure of the taxonomy. The supplementary material to this paper [5] includes a simple and documented example.

[3] We plan to soon provide this feature also in the multi-objective case.

6 Experiments

The improvements obtained by the proposed contributions are evaluated by means of a comprehensive benchmark. To focus on specific results, our expectations are first formulated as research hypotheses. Then, the design of the experimental study is described. In the main part of this section, the hypotheses are checked using statistical testing and the respective observations are discussed.

Page limitations restrict the evaluation to the main hypotheses. The complete source code of the experimental study, tables including all indicator values, convergence plots, as well as empirical attainment surfaces on the bi-objective problems can be found in the supplementary material [5]. In order to exploit the full information provided by the benchmark, we strongly recommend to take this material into account.

6.1 Research Hypotheses

Within this paper, the benchmark results are analyzed with regard to three research hypotheses:

1. MBMO can significantly improve the approximation quality compared to model-free approaches in case of a strictly restricted budget of evaluations.
2. Compared to a single-point proposal, a multi-point proposal can significantly reduce clock time and preparation effort while not significantly deteriorating the results with regard to the budget of evaluations.
3. The structure of mlrMBO implied by the taxonomy allows the realizations of specific steps of the algorithm to be exchanged, benchmarked, and finally improved in a simple and efficient way.

In addition to the new candidate selection methods for the multi-point proposal, the last hypothesis is tested by exchanging the infill criteria for the candidate generation and selection in ParEGO and MSPOT.

6.2 Experimental Setup

Algorithms ParEGO, SMS-EGO, ε-EGO, and MSPOT were implemented using the mlrMBO toolbox. Hence, all different classes of MBMO algorithms (Pareto-, scalarization-, and indicator-based) are covered. As MOEA/D-EGO applies more complex candidate generation and selection phases, and hence would result in additional implementation and space requirements, it is omitted within these experiments. Multi-EGO and MOEA using surrogates are adressed by considering their alternative infill criterion within MSPOT.

The initial design size of the algorithms was set to $n_{\text{init}} = 4d$. Kriging models were fitted with a Matern5/2 kernel. A total budget of $n_{\text{total}} = 40d$ was allowed, resulting in $36d$ points proposed over the iterations. The small n_{init} was chosen intentionally in order to have a high number of sequential evaluations while still operating under a severly restricted total number of evaluations.

In ParEGO, ρ of equation 1 was set to $\rho = 0.05$. The number of uniform steps used for generating the weight vectors was adjusted in a way that approximately $100,000$ weight vectors result in total. The ideal point $\boldsymbol{i}$ was estimated using the minimum objective values of the currently seen observations.

In SMS-EGO, the gap of the additive ε-dominance was estimated using the adaptive formula

$$\boldsymbol{\varepsilon} = \frac{\Delta\mathcal{Y}^*_{\text{approx}}}{|\mathcal{Y}^*_{\text{approx}}| + c \cdot (n_{\text{total}} - n)}, \; \Delta\mathcal{Y}^*_{\text{approx}} = \max(\mathcal{Y}^*_{\text{approx}}) - \min(\mathcal{Y}^*_{\text{approx}}),$$

where n is the current number of evaluations and $c = 1 - 1/(2^m)$ corresponds to the idealized probability of a random solution being non-dominated. min and max are vectorized operations, i. e, the minimum (maximum) for each dimension is returned. $|\mathcal{Y}^*_{\text{approx}}|$ denotes the number of observation in $|\mathcal{Y}^*_{\text{approx}}|$. As reference point for the hypervolume computations, $\boldsymbol{r} = \max(\mathcal{Y}^*_{\text{approx}}) + \mathbf{1}$ was used.

For the evaluation of the first hypothesis, all considered MBMO algorithms are tested against NSGA-II and random search. NSGA-II was taken from the R package MCO and was run with a population size $P = n_{\text{init}}$ for 10 generations. This allows a direct comparison to the MBMO algorithms. As variation operators, simulated binary crossover (SBX) and polynomial mutation (PM) are applied with their standard parameters $p_c = 1$, $\eta_c = 15$, $p_m = \frac{1}{d}$, and $\eta_m = 20$. Random search acts as a baseline. It starts with the same initial design as the MBMO algorithms and randomly proposes the remaining points.

The second hypothesis is analyzed by implementing the candidate generation and selection concepts of section 4 into mlrMBO. The number of points in a batch was set to $N = 4$. To achieve a balanced set of weight vectors in parallel ParEGO, $cN = 20$ ($c = 5$) weight vectors were randomly drawn and reduced using the distance-based filter. As a consequence of the batch evaluation, only $9d$ iterations of the sequential procedure were performed.

As examples for investigating the third hypothesis, also the LCB was considered as infill criterion for optimizing the model of the scalarization within ParEGO. In addition, the multi-objective optimization for generating the candidates in MSPOT was also performed based on the EI and the LCB. As in the direct indicator-based (DIB) MBMO, the factor λ of the LCB was computed based on a given probability level p ($p = 0.5$ in this study) by $\lambda = -\Phi^{-1}(0.5\ \sqrt[m]{p})$.

Due to a full factorial combination of infill criteria and the single- and multi-point candidate selection, in total 4 variants of ParEGO and 6 variants of MSPOT were considered. For SMS-EGO and ε-EGO, one single- and one multi-point variant were assessed, respectively. Hence, 14 MBMO instantiations were benchmarked. For all algorithms, including NSGA-II and random search, 20 runs were performed. All runs with the same index were based on the same initial design, except for NSGA-II which used a random initial population for technical reasons.

The internal single-objective optimization tasks were solved using a focusing random search. It performs large random searches on the decision space, which can be evaluated in parallel to reduce technical overhead when querying the machine learning model, and iteratively shrinks the boundaries of the sample space

Table 2. Test functions designed by combining global optimization problems

Name	d	m	**Internal test functions**
gomop-22	2	2	Branin, 3-Hump-Camel ($\boldsymbol{x} \in [-2, 2]^2$)
gomop-25	2	5	Branin, 3-Hump-Camel ($\boldsymbol{x} \in [-2, 2]^2$), Hartman, Goldstein-Price, 6-Hump-Camel ($x_1 \in [-2, 2]$, $x_2 \in [-1, 1]$)
gomop-52	5	2	Hartman, Rastrigin ($\boldsymbol{x} \in [-0.5, 0.5]^5$])
gomop-55	5	5	Hartman, Rastrigin ($\boldsymbol{x} \in [-0.5, 0.5]^5$), Rosenbrock, Zahkharov ($\boldsymbol{x} \in [-1, 1]^5$), Powell ($\boldsymbol{x} \in [-1, 1]^5$)

around the best obtained point by a factor of 0.5, enforcing local convergence. Additionally, restarts of the whole approach were performed, for a further global optimization effect. In the experiments, a random set of $1,000$ points is evaluated within each of the three focusing steps and three restarts are performed, resulting in total in $9,000$ evaluations for each internal optimization.

For the multi-objective optimization in MSPOT, again the NSGA-II was applied. For the internal optimization, the population size 100 and 90 generations were specified in order to also allow 9000 evaluations of the surrogate models.

Test functions All algorithms were evaluated on 9 test functions. Two settings, ($d = 2$, $d = 5$) and ($m = 2$, $m = 5$), of both, decision and objective space, were considered, respectively. As established test functions, zdt1, zdt2, and zdt3 with $d = 5$ decision and $m = 2$ objective space dimensions, as well as dtlz1 with $d = 5$ and both $m = 2$ and $m = 5$, were used. In addition, the concept of combined multi-objective problems from single-objective problems [17, 20] was utilized in order to design 4 additional test functions. These test functions are based on established global optimization functions and are summarized in Table 2[4]. In order to unify the box constraints of the decision spaces, the respective bounds of each single-objective test function were mapped to $[0, 1]^d$.

Performance Assessment The final Pareto front approximations of the algorithms were compared using three performance indicators: R2, hypervolume, and additive ε [24]. The R2 and the hypervolume indicator were used in their unary variant. Hence, the ε and R2 indicators have to be minimized, whereas the hypervolume has to be maximized.

For each test function, the reference sets for the binary ε-indicator were built from the Pareto-optimal solutions of the union of all available Pareto front approximations. All approximations and reference sets are normalized to the interval $[1, 2]^m$ with respect to the ideal and nadir points given in table 3 before computing the indicators.

All indicators are recommended for performance assessment based on their favorable theoretical properties [24]. As we mainly compare algorithm variants within their respective MBMO class to check our hypotheses, only the metric

[4] For further information: `http://www.sfu.ca/~ssurjano/optimization.html`

Table 3. Nadir and ideal points for each test function

	gomop-22	gomop-25	gomop-52	gomop-55	
Ideal	(0, 0)	(0, -5, 1, 0, -1.1)	(-3.5, 35)	(-3.5, 8.5, 35, 0, 0)	
Nadir	(40, 2.5)	(125, 0, 15, 6, 3.1)	(0, 125)	$(0, 3 \cdot 10^6, 150, 2000, 350)$	
	dtlz2-52	dtlz2-55	zdt1-52	zdt2-52	zdt3-52
Ideal	(0, 0)	(0, 0, 0, 0, 0)	(0, 0)	(0, 0)	(0, -1)
Nadir	(2, 2)	(1.25, 1.25, 1.25, 1.25, 1.25)	(1, 10)	(1, 10)	(1, 10)

corresponding to the internal selection mechanism of the respective MBMO class is shown in the result tables.

6.3 Observations

The results of the experiments are summarized in Tables 4, 5, and 6. Significant improvements ($p = 0.05$) to the baseline algorithms with respect to independent pairwise Wilcoxon tests are indicated by subscripts (r random search, n NSGA-II). In addition, superscripts are added in order to provide information regarding the comparison of the multi-point variants with their original counterpart shown in the left column of each table. $^+$ means no significant deterioration, whereas $^{++}$ corresponds to a significant improvement.

Hypothesis 1 Random search and NSGA-II were outperformed by almost all MBMO algorithms on almost all test functions. The use of kriging models can thus drastically reduce the number of evaluations required to solve multi-objective optimization problems. Surprisingly, the original ParEGO (1-ei) was not able to outperform these baselines on 4 test functions.

Hypothesis 2 The second hypothesis has to be considered separately for the different algorithms. For ε-EGO (cf. Table 4, left), a significant deterioration of the multi-point compared to the single-point variant was observed on only one test function. Hence, the simulated evaluation strategy can be applied to reduce

Table 4. Results of the indicator-based EGO variants with regard to their indicator

	dib-1-eps	dib-4-eps
gomop-22	0.035_{rn}	$\mathbf{0.029}^{+}_{rn}$
gomop-25	$\mathbf{0.074}_{rn}$	0.075^{+}_{rn}
gomop-52	$\mathbf{0.098}_{rn}$	0.121^{+}_{rn}
gomop-55	**0.230**	0.246^{+}
dtlz2-52	$\mathbf{0.003}_{rn}$	0.004_{rn}
dtlz2-55	$\mathbf{0.135}_{rn}$	0.137^{+}_{rn}
zdt1-52	0.024_{rn}	$\mathbf{0.023}^{+}_{rn}$
zdt2-52	0.043_{rn}	$\mathbf{0.038}^{++}_{rn}$
zdt3-52	0.048_{rn}	$\mathbf{0.046}^{+}_{rn}$

dib-1-sms	dib-4-sms
$\mathbf{1.152}_{rn}$	1.136_{rn}
$\mathbf{1.252}_{rn}$	1.235_{rn}
$\mathbf{0.982}_{rn}$	0.959_{rn}
1.169_{rn}	$\mathbf{1.221}^{++}_{rn}$
$\mathbf{1.011}_{rn}$	1.007_{rn}
1.476_{rn}	$\mathbf{1.492}^{++}_{rn}$
$\mathbf{1.171}_{rn}$	1.169_{rn}
$\mathbf{1.133}_{rn}$	1.132_{rn}
$\mathbf{1.105}_{rn}$	1.103^{+}_{rn}

Table 5. Results of the ParEGO variants with regard to the R2 indicator

	1-ei	4-ei	1-lcb	4-lcb
gomop-22	$\mathbf{0.051}_{rn}$	0.051^{+}_{rn}	0.051_{rn}	$\mathbf{0.049}^{+}_{rn}$
gomop-25	0.061	$\mathbf{0.058}^{+}$	0.043_{rn}	$\mathbf{0.043}^{++}_{rn}$
gomop-52	**0.176**	0.177^{+}	$\mathbf{0.103}_{rn}$	0.108^{+}_{rn}
gomop-55	**0.066**	0.068^{+}	$\mathbf{0.042}_{rn}$	0.042^{+}_{rn}
dtlz2-52	0.123	$\mathbf{0.123}^{+}$	$\mathbf{0.110}_{rn}$	0.110_{rn}
dtlz2-55	$\mathbf{0.023}_{rn}$	0.023^{+}_{rn}	$\mathbf{0.024}_{rn}$	0.024^{+}_{rn}
zdt1-52	$\mathbf{0.039}_{rn}$	0.040_{rn}	0.032_{rn}	$\mathbf{0.032}^{+}_{rn}$
zdt2-52	0.052_{rn}	$\mathbf{0.051}^{+}_{rn}$	0.045_{rn}	$\mathbf{0.045}^{+}_{rn}$
zdt3-52	0.070_{rn}	$\mathbf{0.070}^{+}_{rn}$	0.059_{rn}	$\mathbf{0.059}^{+}_{rn}$

clock time and preparation effort without a significant loss of approximation quality. The same holds for the use of multiple weight vectors for generating batch evaluations in ParEGO (cf. Table 5) which did not result in significant deteriorations, except on ZDT1 (ei) and DTLZ2 with $m = 2$ (lcb).

For SMS-EGO (cf. Table 4, right) and MSPOT (cf. Table 6), however, this result could not be confirmed. Only on 2 to 3 of the 9 test functions considered in this study, the multi-point variants were not significantly worse. On two test functions, however, a batch evaluation led to improved results for SMS-EGO.

Hypothesis 3 Also for the third hypothesis, the different MBMO algorithms have to be considered separately. For MSPOT, the exchange of the infill criterion does generally not result in significant performance differences. Only on two of the GOMOP functions, the LCB deteriorates the results compared to mean prediction and EI. Hence, it is possible to exchance specific steps of the algorithm without detoriating the algorithm's performance.

The exchange of the EI and the LCB in ParEGO obtained excellent improvements. On almost all test functions, the results using the LCB are better, sometimes by far margins. The same held for the multi-point variants. Here the taxonomy allowed us to construct a new algorithm variant, that outperforms its original counterpart.

Table 6. Results of the MSPOT variants with regard to the hypervolume indicator

	1-mean	4-mean	1-ei	4-ei	1-lcb	4-lcb
gomop-22	$\mathbf{1.148}_{rn}$	1.142_{rn}	$\mathbf{1.146}_{rn}$	1.141^{+}_{rn}	1.136_{rn}	$\mathbf{1.142}^{+}_{rn}$
gomop-25	$\mathbf{1.246}_{rn}$	1.225_{rn}	$\mathbf{1.245}_{rn}$	1.216_{rn}	$\mathbf{1.248}_{rn}$	1.226_{rn}
gomop-52	$\mathbf{0.907}_{rn}$	0.874_{rn}	$\mathbf{0.908}_{rn}$	0.862_{rn}	$\mathbf{0.904}_{rn}$	0.880_{rn}
gomop-55	$\mathbf{1.145}_{rn}$	1.127^{+}_{rn}	$\mathbf{1.143}_{rn}$	1.124^{+}_{rn}	$\mathbf{1.126}_{rn}$	1.126^{+}_{rn}
dtlz2-52	$\mathbf{1.003}_{rn}$	0.997_{rn}	$\mathbf{1.002}_{rn}$	0.996_{rn}	$\mathbf{1.002}_{rn}$	0.997_{rn}
dtlz2-55	1.414_{rn}	$\mathbf{1.416}^{+}_{rn}$	$\mathbf{1.409}_{rn}$	1.409^{+}_{rn}	1.411_{rn}	$\mathbf{1.414}^{+}_{rn}$
zdt1-52	$\mathbf{1.116}_{rn}$	1.091_{r}	$\mathbf{1.116}_{rn}$	1.094_{r}	$\mathbf{1.115}_{rn}$	1.099_{r}
zdt2-52	$\mathbf{1.057}_{rn}$	1.029_{r}	$\mathbf{1.056}_{rn}$	1.034_{r}	$\mathbf{1.055}_{rn}$	1.029_{r}
zdt3-52	$\mathbf{1.051}_{rn}$	1.022_{r}	$\mathbf{1.054}_{rn}$	1.034_{r}	$\mathbf{1.051}_{rn}$	1.022_{r}

General Recommendations The original one-point ParEGO using the EI performs worse compared to all considered MBMO algorithms. By exchanging the EI with the LCB, however, the approach becomes competetive. The new variant can thus be recommended as a standard choice for the future.

In case of a one-point proposal, SMS-EGO (dib-1-sms) performs better or comparable on almost all test cases. It can be proposed as a general recommendation. If a multi-point proposal is desired, the respective variants of the SMS-EGO (dib-4-sms) and ParEGO (parego-4-lcb) show a comparable performance. As ParEGO only requires a single model, can be parallized without simulated evaluations, and is much faster to compute, in particular on many-objective problems, it can recommended for this case.

6.4 Discussion

The experiments showed two main results: (1) For ParEGO and ε-EGO, no significant deterioration of the results can be observed due to the multi-point proposal; (2) The change from the EI to the LCB significantly improved ParEGO.

Regarding the first result, the infill criteria of ParEGO and ε-EGO still have minor conceptual issues which inhibit the exploitation of the additional information obtained by more frequent updates. ParEGO draws the weight vectors for the scalarization at random. Hence, the implied search directions can point to regions already crowded with observations. By choosing more weight vectors per iteration in a space-filling way, the coverage of the Pareto front is improved. In comparison to MOEA/D-EGO, which evaluates all weight vectors in each generation and chooses based on the maximum EI values of a predefined, fixed clustering, the proposed procedure does not suffer from a systematic bias towards certain regions [7, 20]. In ε-EGO, the optimization of an indicator based on two sets is reduced to one based on a set and a single solution. This may hinder the finetuning of $\mathcal{Y}^*_{\text{approx}}$ with regard to the global indicator.

Main result (2) can be caused by the properties of the fitness landscape implied by the EI. It has plateaus whose size increases with decreasing uncertainty of the model. The maxima of the EI lie within small basins surrounded by these plateaus. They are hard to find for both local optimization algorithms and global sampling strategies, such as the focusing random search. In particular, if a weight vector pointing to crowded region is selected, the EI can show values far below 10^{-6}, even after only 2-3 iterations. By switching to the LCB, a global trend is available which can be exploited during the internal optimization.

7 Conclusions and Outlook

In this paper, a taxonomy for MBMO algorithms was presented for the first time. Based on this taxonomy, an R toolbox was designed and some established MBMO algorithms were implemented. In order to allow batch processing, the candidate generation step of all considered algorithms was enhanced to a multi-point proposal. In addition, the internal infill and optimization criteria were

exchanged and different variants of the MBMO algorithms were compared within a comprehensive benchmark.

For ParEGO and ε-EGO, the multi-point variants did not significantly deteriorate the results. They even improved the approximation quality in some cases. Moreover, the change from the EI to the LCB could improve the results of the internal optimization within ParEGO.

In future work, the scalability of the multi-point proposal with the batch size N has to be further evaluated. Moreover, systematic problems, such as the random choice of the weight vector in ParEGO, should be tackled. A simple strategy would be to redraw a weight vector in case of too low EI values. In addition, different shifts of the ideal (ParEGO) or reference point (SMS-EGO) can be used for constructing different subproblems for multi-point proposals. The simulated evaluation strategy used in the DIBs can be combined with fake observations and a refit of the model in order to improve exploration or exploitation of certain regions. The framework created by the taxonomy and the R toolbox make this possible in a structured and convenient way.

References

1. Bautista, D.C.: A Sequential Design For Approximating The Pareto Front Using the Expected Pareto Improvement Function. Ph.D. thesis, Ohio State University (2009)
2. Binois, M., Ginsbourger, D., Roustant, O.: Quantifying uncertainty on pareto fronts with gaussian process conditional simulations. European Journal of Operational Research (accepted, available online), 1–9 (2014)
3. Bischl, B., Wessing, S., Bauer, N., Friedrichs, K., Weihs, C.: MOI-MBO: Multiobjective infill for parallel model-based optimization. In: Learning and Intelligent Optimization Conference (LION8). pp. 173–186 (2014)
4. Bischl, B., Horn, D., Bossek, J., Richter, J., Lang, M.: Package mlrMBO (2014), `https://github.com/berndbischl/mlrMBO`
5. Bischl, B., Horn, D., Wagner, T.: Model-Based Multi-Objective Optimization: Taxonomy, Multi-Point Proposal, Toolbox and Benchmark: Supplementary Material (2014), `http://dx.doi.org/10.6084/m9.figshare.1275926`
6. Bischl, B., Lang, M., Bossek, J., Judt, L., Richter, J., Kuehn, T., Studerus, E.: Package mlr: Machine Learning in R (2014), `http://cran.r-project.org/web/packages/mlr/index.html`
7. Brockhoff, D., Wagner, T., Trautmann, H.: On the properties of the R2 indicator. In: Soule, T., et al. (eds.) Proc. 2012 Genetic and Evolutionary Computation Conference (GECCO 2012), Philadelphia, US. pp. 465–472 (2012)
8. Couckuyt, I., Deschrijver, D., Dhaene, T.: Fast calculation of multiobjective probability of improvement and expected improvement criteria for pareto optimization. Journal of Global Optimization (accepted, available online), 1–21 (2014)
9. Emmerich, M.T.M., Deutz, A., Klinkenberg, J.W.: Hypervolume-based expected improvement: Monotonicity properties and exact computation. In: Proc. IEEE Congress on Evolutionary Computation (CEC). pp. 2147–2154 (2011)
10. Janusevskis, J., Riche, R.L., Ginsbourger, D., Girdziusas, R.: Expected improvements for the asynchronous parallel global optimization of expensive functions:

Potentials and challenges. In: Hamadi, Y., et al. (eds.) Learning and Intelligent Optimization - 6th International Conference, (LION). pp. 413–418. Springer (2012)
11. Jeong, S., Obayashi, S.: Efficient global optimization (EGO) for multi-objective problem and data mining. In: Corne, D., et al. (eds.) Proc. IEEE Congress on Evolutionary Computation (CEC 2005). pp. 2138–2145. IEEE (2005)
12. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization 21(4), 345–383 (2001)
13. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization 13(4), 455–492 (1998)
14. Keane, A.J.: Statistical improvement criteria for use in multiobjective design optimization. AIAA Journal 44(4), 879–891 (2006)
15. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 10(1), 50–66 (2006)
16. Knowles, J., Nakayama, H.: Meta-modeling in multiobjective optimization. In: Branke, J., et al. (eds.) Multiobjective Optimization - Interactive and Evolutionary Approaches, pp. 245–284. Springer, Berlin (2008)
17. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: Yao, X., et al. (eds.) Proc. 8th Int'l Conf. Parallel Problem Solving from Nature (PPSN). pp. 792–802. Springer, Berlin (2004)
18. Ponweiser, W., Wagner, T., Biermann, D., Vincze, M.: Multiobjective optimization on a limited amount of evaluations using model-assisted $\mathcal{S}$-metric selection. In: Rudolph, G., et al. (eds.) Proc. 10th Int'l Conf. Parallel Problem Solving from Nature (PPSN). pp. 784–794. Springer, Berlin (2008)
19. Voutchkov, I., Keane, A.J.: Multiobjective optimization using surrogates. In: Proc. 7th Int'l. Conf. Adaptive Computing in Design and Manufacture (ACDM), Bristol, UK. pp. 167–175 (2006)
20. Wagner, T.: Planning and Multi-Objective Optimization of Manufacturing Processes by Means of Empirical Surrogate Models. Vulkan Verlag, Essen (2013)
21. Wagner, T., Emmerich, M., Deutz, A., Ponweiser, W.: On expected-improvement criteria for model-based multi-objective optimization. In: Schaefer, R., et al. (eds.) Proc. 11th Int'l. Conf. Parallel Problem Solving From Nature (PPSN). pp. 718–727. Springer, Berlin (2010)
22. Zaefferer, M., Bartz-Beielstein, T., Naujoks, B., Wagner, T., Emmerich, M.: A case study on multi-criteria optimization of an event detection software under limited budgets. In: Purshouse, R., et al. (eds.) Proc. 7th Int'l. Conf. Evolutionary Multi-Criterion Optimization (EMO). pp. 756–770. Springer, Berlin (2013)
23. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with gaussian process model. IEEE Transactions on Evolutionary Computation 4(3), 456–474 (2010)
24. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: An analysis and review. Transactions on Evolutionary Computation 8(2), 117–132 (2003)