Optimally Weighted Ensembles for Efficient Multi-Objective Optimization

Gideon Hanse[1], Roy de Winter[1], Bas van Stein[1], and Thomas Bäck[1]

Leiden Institute of Advanced Computer Science, Leiden, The Netherlands
i.j.g.hanse@umail.leidenuniv.nl,
{r.de.winter,b.van.stein,t.h.w.baeck}@liacs.leidenuniv.nl

Abstract. The process of industrial design engineering is often involved with the simultaneous optimization of multiple expensive objectives. The surrogate assisted multi-objective S-Metric Selection, Efficient Global Optimization (SMS-EGO) algorithm is one of the more popular algorithms to solve this kind of problems. We propose an extension of the SMS-EGO algorithm with optimally weighted, linearly combined ensembles of regression models to improve its objective modelling capabilities. Multiple (different) surrogates are combined into one optimally weighted ensemble per objective using a model agnostic uncertainty quantification method to balance between exploration and exploitation. The performance of the proposed algorithm is evaluated on a diverse set of benchmark problems with a small budget of 25 evaluations of the real objective functions. The results show that the proposed Ensemble-based S-Metric Selection Efficient Global Optimization (E-SMS-EGO) algorithm outperforms the state-of-the-art algorithms in terms of efficiency, robustness and spread across the objective space.

Keywords: Multi-objective Optimization · Efficient Global Optimization · Surrogate Models · Ensemble Models · Uncertainty Quantification · S-Metric Selection · Industrial Design.

1 Introduction

The process of industrial design engineering is often involved with the optimization of multiple highly costly objective functions [17], which can be formulated as a Multi-objective Optimization Problem (MOP):

$$\min_{\vec{x}} \vec{f}(\vec{x}) \text{ , where } \vec{f}(\vec{x}) = [f_1(\vec{x}), \dots, f_m(\vec{x})] \tag{1.1}$$

in which $\vec{f}$ is a collection of m objective functions, where $\vec{x} = [x_1, \dots, x_n]$ is a solution on n independent variables in the feasible region $\Omega \subseteq \mathbb{R}^n$. In a multi-objective optimization setting, different objectives are commonly conflicting with each other, where one objective can not be improved without deteriorating on another objective function. Consequently, instead of finding one single solution, the goal is to find a collection Pareto-optimal solutions $\mathcal{P}$ [13]:

$$\mathcal{P} := \{\vec{x} \in \Omega \mid \nexists \vec{x}' \in \Omega : \vec{f}(\vec{x}') \preceq \vec{f}(\vec{x})\} \tag{1.2}$$

where $\preceq$ indicates Pareto-dominance. A solution $\vec{x}$ dominates another solution $\vec{x}'$ if and only if

$$\forall_i (f_i(\vec{x}) \leq f_i(\vec{x}')) \text{ and } \exists f_i(\vec{x}) < f_i(\vec{x}'), \quad i = 1, \ldots, m \tag{1.3}$$

The set of solutions $\mathcal{P}$ together form a *Pareto-front* in the m-dimensional objective landscape. Objective functions in case of industrial design are often unknown, requiring an iterative evaluation process of proposed design configurations by running simulations [14,31] or by building prototypes [19]. Such evaluations are often extremely costly and time-consuming, even though more and more computational power becomes available [17].

To avoid spending an excessive amount of time and resources at design evaluation, a widely used method is to approximate real objective functions using surrogate models [19]. In particular, ensembles of multiple surrogate models have been successfully applied to approximate costly objective functions and they were shown to yield great performance in optimization tasks [29,19].

Unfortunately, existing techniques for multi-objective optimization often still depend on domain-specific prior knowledge about the given optimization tasks [18]. To eliminate this need for prior understanding of MOP landscapes, we propose the *Ensemble-based S-Metric Selection Efficient Global Optimization* (E-SMS-EGO) algorithm, which combines surrogate models into adaptive ensembles to solve computationally very expensive multi-objective optimization problems in an efficient manner.

2 Related Work

A large body of academic work has already been dedicated to study ensembles of surrogates, as well as multi-objective optimization problems. In the following, we cover the most relevant approaches that lie at the foundation of the proposed algorithm.

2.1 Ensembles of Surrogate Models

Combining the output of multiple surrogate models into an ensemble has repeatedly been shown to be beneficial to optimization processes in both practical applications and artificial test settings. For instance, ensembles of surrogate models have improved the optimization of wind turbine allocation [37], the minimization of car crash impact in car designs [2] and more [19,35].

Aside from classical ensemble techniques, e.g., Ranking, Bagging and Boosting [19], a well-proven method to generate ensembles is by computing the weighted average of multiple surrogate models [16]. Multiple frameworks to find optimal weights have been proposed so far, for which the performance depends heavily on the nature of the given optimization tasks [19].

When generating such weighted ensembles, weights are mainly assigned based on the contributions of individual models [15]. In general, surrogate models that perform better are given higher weights, and the weights for the worst performing models are reduced to zero. As the weights are based on the individual model performance, the composition of the found ensembles strongly depends on the choice of performance metric [33,15].

In literature, weighted ensemble methods are roughly divided into two categories: Globally weighted averaging, and locally weighted averaging of models [19].

In globally weighted averaging methods, the complete design space is considered altogether in the calculation of individual model performances, and the outputs of the models are combined using the same weight across the whole input space. As a first attempt at weighted model combination, Goel et al. [16] combined several regression models into ensembles by globally weighting the models based on their performances to approximate expensive objective functions. Since then, the main framework from this study has been improved upon in multiple ways, e.g., improving efficiency by clustering the design space [36], introducing an optimization procedure for finding optimal weights [2], and by using a covariance matrix of prediction errors to efficiently find weights [29,33].

Friese et al. [15] showed that any ensemble with positive weights can, in terms of measured error, never perform worse than the worst performing base model, and always has a chance to perform better than any of the individual base models due to the convex nature of the weight combination. Moreover, they proposed to perform an evolutionary search over model weights to scale up the number of included base models.

In addition, more sophisticated approaches have been studied to generate ensembles based on local accuracy measures. For example, Acar [1] used the cross-validated prediction variance as a local accuracy measure to indicate individual model performance, while still providing fixed weights over the entire input space. Other approaches also assign model weights differently across the input space [19]. By doing so, ensembles are better capable of capturing local trends in specified regions of the design space [37].

2.2 Efficient Global Optimization

In general, a small number of initial evaluated data points is not sufficient to obtain a good representation of the overall objective landscape. Therefore, new points have to be sampled to increase the predictive accuracy of the surrogate models.

A widely used algorithm for the sequential optimization of expensive black-box functions is the *Efficient Global Optimization* (EGO) algorithm as introduced by Jones et al. [20]. The EGO algorithm heavily exploits the proposed surrogate model by sequentially choosing new candidate points for evaluation. These points are chosen based on the prediction of the model, as well as the uncertainty about the prediction at that point by introducing an *infill-criterion*. By addressing both the prediction as well as the uncertainty, the EGO algorithm autonomously balances between exploration and exploitation of the search space.

2.3 Uncertainty Quantification

In the EGO framework, the uncertainty of point predictions has to be taken into account when predicting objective functions, e.g. in terms of variances, standard deviations, or confidence intervals. In such cases, an infill criterion is used as a combined metric of the predicted value and the uncertainty of the prediction, e.g., *Expected Improvement* [20], *Lower Confidence Bound* [8], and *Probability of Improvement* [32].

Some regression models automatically address the confidence of predictions as they also provide an estimation of the prediction variance [22,19].

However, the majority of regression models is not equipped with such built-in variance estimation properties, which calls for an external *uncertainty quantification* (UQ) measure in order to be adopted to the EGO framework. Van Stein et al. [30] provide a fine UQ measure as such that is independent of surrogate modeling assumptions by addressing the empirical prediction error at a given point, as well as the variability of the k nearest neighbours based on the euclidean distance to these neighbouring points. This allows for combining different surrogate models in the EGO framework.

2.4 Model-Based Multi-Objective Optimization

In the multi-objective optimization setting, surrogate models have been used for a wide variety of tasks [3]. For example, Loshchilov et al. [25], Bandara et al. [5] trained surrogate models to distinguish dominated solutions from non-dominated solutions. Additionaly, surrogate models have been used to approximate the increase in hypervolume of new proposed individuals [4]. However, as described earlier, the focus of the present study is to apply surrogate models for the approximation of multiple objective functions. Methods that do so are generally scalarization-based [27,21,38], Pareto-based [10,9,24] or Direct Indicator-based [28,34]. A very well known Direct Indicator-based method is the model-assisted S-Metric selection approach by Ponweiser et al. [28], which accurately identifies promising data points by optimizing the amount of added hypervolume. Even though the literature about surrogate ensembles and model-based multi-objective optimization is quite extensive, the question on how to combine the two topics in a knowledgeable manner has, to the best of our knowledge, rarely been addressed so far.

3 E-SMS-EGO

In this paper, we propose the *E-SMS-EGO* algorithm, extending the *SMS-EGO* algorithm [28] with optimally weighted ensembles, combined with uncertainty quantification in order to efficiently solve MOPs.

3.1 Initial Sampling

An initial set of data points is obtained with the Latin Hypercube Sampling (LHS) method [26], which provides an equally distributed sample of data points across the search domain. LHS ensures that the amount of information that the surrogate models can derive from the sample is maximized. The initial data points are evaluated on the objective functions to obtain the corresponding objective values for the initial data set.

3.2 Finding Optimal Ensemble Weights

Subsequently, a well-performing and robust ensemble is generated for every objective function by finding the optimal linear combination of weights per objective function. Currently, ensembles are created by calculating the weighted average of five base models, i.e. a kriging model, radial basis function, decision tree, support vector machine and a multivariate adaptive regression spline. However, as the method is model-agnostic, note that it can be used with any number of regression models of any kind.

In *E-SMS-EGO*, the optimal weights are found per objective by means of 10-fold cross validation, largely based on the linear combination method proposed by Friese et al. [15]. First, all possible weight combinations are obtained by calculating p possible integer partitions with size k (the amount of base models) out of the integer 10. Dividing these partitions by 10 results in weight matrix W:

$$W_{p,k} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0_{1,k} \\ 0.9 & 0.1 & 0 & \cdots & 0 & 0_{2,k} \\ 0.8 & 0.1 & 0.1 & \cdots & 0 & 0_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0.1 & 0.9_{p-1,k} \\ 0_{p,1} & 0_{p,2} & 0_{p,3} & \cdots & 0_{p,k-1} & 1_{p,k} \end{pmatrix}$$

where: p: number of possible weight combinations. k: amount of base models.

Out of this collection of possible weight combinations, the optimal weights are found separately for all of the objective functions by executing the following steps according to the 10-fold cross validation procedure:

1. First, the base models are trained on the training partition of the cross-validation fold.
2. Subsequently, the trained models are fitted separately to predict the objective value of the configurations in the test partition, resulting in a prediction matrix of size k times the amount of test points in the fold.
3. Next, an ensemble prediction is determined by calculating the weighted average using every possible weight combination, i.e., the rows in matrix W, as in Equation 3.2.
4. Finally, the prediction error of the ensemble predictions are calculated.

This results in ten MSE scores for all of the possible weight combinations, which are averaged to get the cross-validated MSE score per weight vector. As a result, the combination of weights with the minimal corresponding cross-validated

MSE value is selected as optimal weight vector $\vec{w}^*_{obj}$ to create the ensemble for approximating the objective function in question.

3.3 Minimizing Ensemble Predictions

Subsequently, a set of potential solutions is found by optimizing the $k-NN$ Ensemble Prediction (KPV) infill criterion per objective:

$$KPV = EPV(\vec{x}) - \widehat{U}_{k-NN} \tag{3.1}$$

consisting of the Ensemble Predicted Value (EPV):

$$EPV(\vec{x}) = \sum_{i=0}^{N} \vec{w}_i \cdot \hat{f}_i(\vec{x})^T, \text{with} \tag{3.2}$$

$\hat{f}_i$: an individual base model.
$\vec{w}_i$: the vector of best weights of the corresponding objective function.

and the U_{k-NN} measure for the uncertainty about the prediction as introduced by van Stein et al. [30]:

$$\widehat{U}_{k-NN} = \frac{\sum\limits_{i \in N(\vec{x})} w_i^k \left| EPV(\vec{x}) - y_i \right|}{\sum\limits_{i \in N(\vec{x})} w_i^k} + \frac{\min\limits_{i \in N(\vec{x})} d(\vec{x}_i, \vec{x})}{\max\limits_{\vec{x}_i, \vec{x}_j \in \chi} d(\vec{x}_i, \vec{x})} \hat{\sigma}, \text{with} \tag{3.3}$$

$$w_i = 1 - \frac{d(\vec{x}_i,\vec{x})}{\sum\limits_{i \in N(\vec{x})} d(\vec{x}_i,\vec{x})}, \hat{\sigma} = \sqrt{\mathrm{Var}\left[\{y_i\}_{i \in N(\vec{x})} \cup \{\hat{f}(\vec{x})\} \right]}.$$

Multiple experiments were conducted to investigate the efficacy of using optimally weighted ensembles in combination with uncertainty quantification, which led to the implementation of the $k-NN$ infill criterion. Supplementary material can be found at `https://github.com/Gitdeon/E-SMS-EGO`.

3.4 *S*-Metric Selection

The minimization process in step 3.3 is repeated multiple times, such that a collection of potential points in the search domain is obtained for all of the objectives in parallel. These points are evaluated on all of the composed ensembles, resulting in multiple predictions that can be used to estimate a Hypervolume score for all potential points. The greatest contributor is then found and selected for evaluation with the S-Metric selection approach as described in Section 2.4.

4 Experiments and Results

The E-SMS-EGO algorithm is compared to state-of-the-art algorithms in terms of Hypervolume and spread of generated solutions.

The competing algorithms are *NSGA-II* [10], *MOEA/D*[38] and *C-TAEA*[24]. To ensure a fair comparison, we used LHS to obtain the same initial sample for all of the algorithms, with a size of $5 \times N$, with N : number of input variables. As the goal is to acquire a well-spread Pareto-front in as little function evaluations as possible, the iteration budget on top of the initial sample was limited to 25 evaluations. Since the competing algorithms make use of populations instead of a single point per iteration, the population sizes and number of generations were both set to 5, also resulting in a budget of 25 evaluations. The shown results for the different algorithms are averages over ten runs with different initial samples and random seeds, guaranteeing reliability of the results.

In order to compare the algorithms, a diverse collection of multi-objective optimization problems was composed, including some artificially designed two-objective problems and real-world like problems. An overview of the test problems is provided in Table 4.1, showing input dimension (n), Lower Bounds (LB) and Upper Bounds (UB) of the input variables, number of objectives (k) and hypervolume reference point (ref). Problems are artificially designed (AD) or real-world like (RWL).

Table 4.1. Artificially designed and Real World Like multi-objective optimization problems as implemented in the *pymoo* package [6].

Problem	type	k	n	LB	UB	ref
BNH [7]	AD	2	2	[0, 0]	[5, 3]	[140, 50]
TNK [12]	AD	2	2	[0, 0]	[π, π]	[2, 2]
CTP1 [11]	AD	2	2	[0, 0]	[1, 1]	[1, 2]
ZDT4 [12]	AD	2	10	$[0, -5, \ldots, -5]^n$	$[1, 5, \ldots, 5]^n$	[1,260]
KSW [23]	AD	2	3	[-5, -5, -5]	[5, 5, 5]	[-10, 2]
WB [17]	RWL	2	4	[0.125, 0.1, 0.1, 0.125]	[5, 10, 10, 5]	[350,1]
CSI [9]	RWL	3	7	[0.5, 0.45, 0.5, 0.5, 0.875, 0.4, 0.4]	[1.5, 1.35, 1.5, 1.5, 2.625, 1.2, 1.2]	[42, 4.5, 13]

4.1 Results

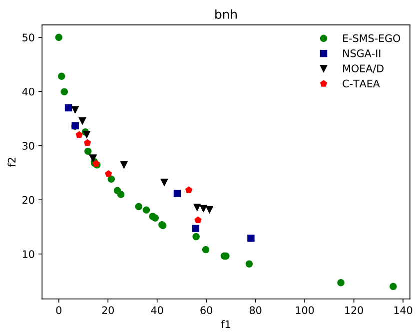
As becomes clear in Table 4.2, E-SMS-EGO significantly outperforms NSGA-II, MOEA/D and C-TAEA in terms of the Hypervolume scores of the obtained Pareto-fronts. Also, in most cases, the found standard deviation in Hypervolume score is lower for E-SMS-EGO, suggesting that the proposed method is more robust and stable than the competing algorithms. On some functions, e.g. TNK, WB, some of the competing algorithms show very poor results in terms of Hypervolume, with abnormally high standard deviations. In these cases, the algorithms did not succeed to find enough feasible, Pareto-optimal solutions below the reference point, therefore receiving a Hypervolume score of 0 in some of the runs. On some problems, especially MOEA/D seemed to perform poorly with a small evaluation budget. However, E-SMS-EGO did not seem to suffer from this issue and was well able to find a Pareto-front in all of the runs.

Table 4.2. Mean Hypervolume score with respect to the reference point for each test function. The best results per test function are shown in boldface if they were significantly higher according to Welch's t-test with $\alpha : 0.05$.

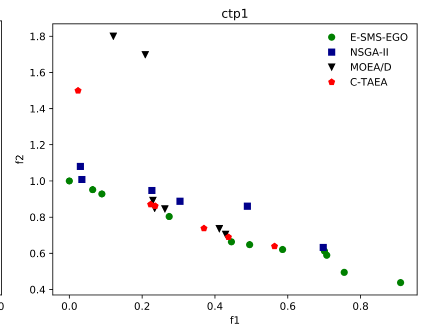
Problem	Measure	NSGA-II	MOAE/D	C-TAEA	E-SMS-EGO
BNH	HV	4760	4617	4723	**5035**
	Std.	134.4	209.8	157.7	35.85
TNK	HV	1.849	0.000	1.015	**3.926**
	Std.	0.597	0.000	1.134	0.116
CTP1	HV	1.115	1.136	1.103	**1.261**
	Std.	0.088	0.061	0.085	0.016
ZDT4	HV	162.1	116.5	161.0	**176.0**
	Std.	18.90	43.16	17.71	14.62
KSW	HV	41.47	43.04	42.01	**56.09**
	Std.	15.05	19.35	12.97	9.923
WB	HV	32.90	1.217	32.49	**33.63**
	Std.	1.298	5.969	4.690	1.797
CSI	HV	11.14	10.56	16.67	**19.41**
	Std.	3.205	2.092	1.101	0.260

In addition, Figures 4.1 and 4.2 show the Pareto-frontiers obtained by the four algorithms on five of the test functions. Here, it is observed that E-SMS-EGO in general succeeds to find the best Pareto-fronts compared to the competing algorithms, as the solutions are located more towards the minimal values on all objectives. In addition to finding more Pareto-optimal solutions, the solutions found by E-SMS-EGO are well-spread across the objective space, which is demonstrated nicely, especially for the BNH, CTP1, KSW and CSI problems.

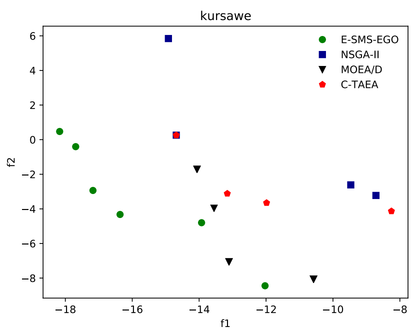
Furthermore, it is shown that, for some problems, only a small number of Pareto-optimal solutions could be found, which is most likely explained by the limited number of allowed function evaluations. Especially for NSGA-II and MOEA/D, which were only allowed small population sizes, this explains why so little Pareto-optimal solutions were found. However, the vast majority of solutions that were found by the competing algorithms are still inferior to the solutions found by E-SMS-EGO, verifying that the proposed method beats the competing algorithms altogether in terms of efficacy in multi-objective optimization.



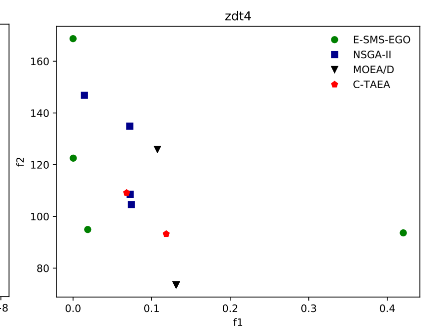
(a) Pareto-fronts BNH



(b) Pareto-fronts CTP1



(c) Pareto-fronts KSW



(d) Pareto-fronts ZDT4

Fig. 4.1. Pareto frontiers obtained by the four algorithms on four of the test functions.

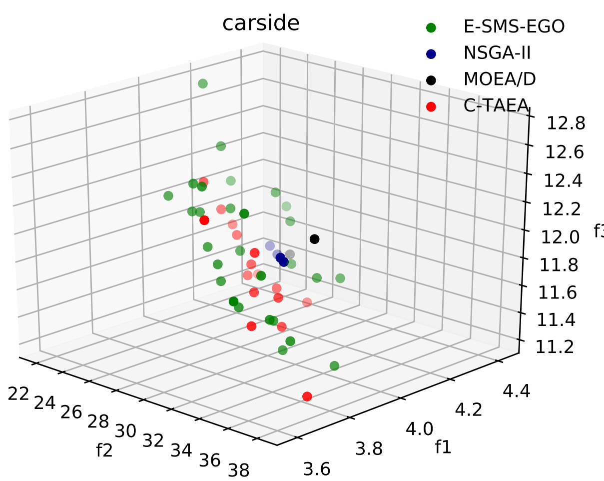


Fig. 4.2. Pareto frontiers obtained by the four algorithms on the three-objective CSI problem.

5 Conclusions and Future Work

In this paper, the novel Ensemble-based Efficient Global Optimization $S-$Metric Selection (E-SMS-EGO) algorithm is proposed, and has been shown to be successful in finding well-performing, Pareto-optimal solutions to multi-objective optimization problems with a limited evaluation budget. By heavily exploiting already known data points, E-SMS-EGO has been shown to outperform comparable state-of-the art multi-objective optimization algorithms, i.e., NSGA-II, MOEA/D and C-TAEA on a diverse collection of artificially designed and real world like test problems.

Multiple experiments were performed with different techniques to compose the proposed algorithm in an optimal way. This resulted in an algorithm that improves upon the SMS-EGO algorithm by using optimally weighted ensembles of regression models as surrogates. By further extending the algorithm with the k-NN variance measure as a method of uncertainty quantification, E-SMS-EGO was able to find minimal solutions that were nicely spread across the objective space, with just a small number of function evaluations.

The algorithm can be adapted in several ways which might boost the performance in the future e.g. local ensemble weighting methods might improve performance, more than five base models could be incorporated into the ensembles, hyper-parameter optimization of base models can still be done, other multi-objective infill criteria can be considered, and finally a constraint handling mechanism could be in cooperated in the future.

References

1. Acar, E.: Various approaches for constructing an ensemble of metamodels using local measures. Structural and Multidisciplinary Optimization **42**, 879–896 (12 2010). https://doi.org/10.1007/s00158-010-0520-z
2. Acar, E., Rais-Rohani, M.: Ensemble of metamodels with optimized weight factors. Structural and Multidisciplinary Optimization **37**, 279–294 (04 2008). https://doi.org/10.1007/s00158-008-0230-y
3. Allmendinger, R., Emmerich, M., Hakanen, J., Jin, Y., Rigoni, E.: Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case. Journal of Multi-Criteria Decision Analysis **24**, 5–24 (12 2016). https://doi.org/10.1002/mcda.1605
4. Azzouz, N., Bechikh, S., Ben Said, L.: Steady state ibea assisted by mlp neural networks for expensive multi-objective optimization problems. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. p. 581–588. GECCO '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2576768.2598271
5. Bandaru, S., Ng, A.H.C., Deb, K.: On the performance of classification algorithms for learning pareto-dominance relations. In: 2014 IEEE Congress on Evolutionary Computation (CEC). pp. 1139–1146 (2014). https://doi.org/10.1109/CEC.2014.6900641
6. Blank, J., Deb, K.: Pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
7. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag, Berlin, Heidelberg (2006)

8. Cox, D.D., John, S.: Sdo: A statistical method for global optimization. In: in Multidisciplinary Design Optimization: State-of-the-Art. pp. 315–329 (1997)
9. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
11. Deb, K.: Multiobjective Optimization Using Evolutionary Algorithms. Wiley, New York (01 2001)
12. Deb, K., Pratap, A., Meyarivan, T.: Constrained test problems for multi-objective evolutionary optimization. vol. 1993, pp. 284–298 (03 2001). https://doi.org/10.1007/3-540-44719-9_20
13. Ehrgott, M.: Vilfredo Pareto and multi-objective optimization, pp. 447–453. Documenta mathematica (2012)
14. Fletcher, C.A.J.: Computational Fluid Dynamics: An Introduction. Springer Berlin Heidelberg, Berlin, Heidelberg (1988). https://doi.org/10.1007/978-3-642-97035-1_1
15. Friese, M., Bartz-Beielstein, T., Emmerich, M.: Building ensembles of surrogates by optimal convex combination. Bioinspired Optimization Methods and their Applications pp. 131–143 (05 2016)
16. Goel, T., Haftka, R., Shyy, W., Queipo, N.: Ensemble of surrogates. Structural and Multidisciplinary Optimization **33**, 199–216 (03 2007). https://doi.org/10.1007/s00158-006-0051-9
17. Gong, W., Cai, Z., Zhu, L.: An efficient multiobjective differential evolution algorithm for engineering design. Structural and Multidisciplinary Optimization **38**, 137–157 (04 2009). https://doi.org/10.1007/s00158-008-0269-9
18. Horn, D., Wagner, T., Biermann, D., Weihs, C., Bischl, B.: Model-based multi-objective optimization: Taxonomy, multi-point proposal, toolbox and benchmark. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) Evolutionary Multi-Criterion Optimization. pp. 64–78. Springer International Publishing, Cham (2015)
19. Jiang, P., Zhou, Q., Shao, X.: Surrogate-Model-Based Design and Optimization (01 2020). https://doi.org/10.1007/978-981-15-0731-1_7
20. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**, 455–492 (12 1998). https://doi.org/10.1023/A:1008306431147
21. 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)
22. Krige, D.: A statistical approach to some basic mine valuation problems on the witwatersrand, by d.g. krige, published in the journal, december 1951 : introduction by the author. Journal of the Southern African Institute of Mining and Metallurgy **52**(9), 201–203 (1952). https://doi.org/10.10520/AJA0038223X_4858
23. Kursawe, F.: A variant of evolution strategies for vector optimization. In: Proceedings of the 1st Workshop on Parallel Problem Solving from Nature. p. 193–197. PPSN I, Springer-Verlag, Berlin, Heidelberg (1990)
24. Li, K., Chen, R., Fu, G., Yao, X.: Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Transactions on Evolutionary Computation **23**(2), 303–315 (2019)
25. Loshchilov, I., Schoenauer, M., Sebag, M.: Comparison-based optimizers need comparison-based surrogates. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) Parallel Problem Solving from Nature, PPSN XI. pp. 364–373. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

26. Mckay, M., Beckman, R., Conover, W.: A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. Technometrics **21**, 239–245 (05 1979). https://doi.org/10.1080/00401706.1979.10489755
27. Miettinen, K., Mäkelä, M.: On scalarizing functions in multiobjective optimization. OR Spectrum **24**, 193–213 (01 2002). https://doi.org/10.1007/s00291-001-0092-9
28. Ponweiser, W., Wagner, T., Biermann, D., Vincze, M.: Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) Parallel Problem Solving from Nature – PPSN X. pp. 784–794. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
29. Shi, R., Liu, L., Long, T., Liu, J.: An efficient ensemble of radial basis functions method based on quadratic programming. Engineering Optimization **48**(7), 1202–1225 (2016). https://doi.org/10.1080/0305215X.2015.1100470
30. van Stein, B., Wang, H., Kowalczyk, W., Bäck, T.: A novel uncertainty quantification method for efficient global optimization. In: Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications. pp. 480–491. Springer International Publishing, Cham (2018)
31. Szabó, B., Babuška, I.: Introduction to Finite Element Analysis: Formulation, Verification and Validation. Wiley Series in Computational Mechanics, Wiley (2011), `https://books.google.nl/books?id=bbi7cQAACAAJ`
32. Ulmer, H., Streichert, F., Zell, A.: Evolution strategies assisted by gaussian processes with improved preselection criterion. In: The 2003 Congress on Evolutionary Computation, 2003. CEC '03. vol. 1, pp. 692–699 Vol.1 (2003). https://doi.org/10.1109/CEC.2003.1299643
33. Viana, F., Haftka, R., Steffen, Jr, V.: Multiple surrogates: How cross-validation errors can help us to obtain the best predictor. Structural and Multidisciplinary Optimization **39**, 439–457 (10 2009). https://doi.org/10.1007/s00158-008-0338-0
34. Wagner, T.: Planning and Multi-Objective Optimization of Manufacturing Processes by Means of Empirical Surrogate Models. Vulkan Verlag, Essen (2013)
35. Ye, P.: A review on surrogate-based global optimization methods for computationally expensive functions. vol. 7, pp. 68–84 (2019)
36. Ye, P., Pan, G.: Global optimization method using ensemble of metamodels based on fuzzy clustering for design space reduction. Engineering with Computers **33**, 573–585 (10 2017). https://doi.org/10.1007/s00366-016-0490-x
37. Zhang, J., Chowdhury, S., Messac, A.: An adaptive hybrid surrogate model. Struct. Multidiscip. Optim. **46**(2), 223–238 (Aug 2012). https://doi.org/10.1007/s00158-012-0764-x
38. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Transactions on Evolutionary Computation **14**(3), 456–474 (2009)