Multi-Objective Optimization with Controlled Model Assisted Evolution Strategies

Jan Braun jan.braun@tu-dortmund.de
Institute for Control and Systems Engineering, TU Dortmund,
Dortmund, 44221, Germany

Johannes Krettek johannes.krettek@tu-dortmund.de
Institute for Control and Systems Engineering, TU Dortmund,
Dortmund, 44221, Germany

Frank Hoffmann frank.hoffmann@tu-dortmund.de
Institute for Control and Systems Engineering, TU Dortmund,
Dortmund, 44221, Germany

Torsten Bertram torsten.bertram@tu-dortmund.de
Chair for Control and Systems Engineering, TU Dortmund,
Dortmund, 44221, Germany

Abstract

Evolutionary algorithms perform robust search in complex and high dimensional search spaces, but require a large number of fitness evaluations to approximate optimal solutions. These characteristics limit their potential for hardware in the loop optimization and problems that require extensive simulations and calculations. Evolutionary algorithms do not maintain their knowledge about the fitness function as they only store solutions of the current generation. In contrast, model assisted evolutionary algorithms utilize the information contained in previously evaluated solutions in terms of a data based model. The convergence of the evolutionary algorithm is improved as some selection decisions rely on the model rather than to invoke expensive evaluations of the true fitness function. The novelty of our scheme stems from the preselection of solutions based on an instance based fitness model, in which the selection pressure is adjusted to the quality of model. This so-called λ-control adapts the number of true fitness evaluations to the monitored model quality. Our method extends the previous approaches for model assisted scalar optimization to multi-objective problems by a proper redefinition of model quality and preselection pressure control. The analysis on multi-objective benchmark optimization problems not only confirms the superior convergence of the model assisted evolution strategy in comparison with a multi-objective evolution strategy but also the positive effect of regulated preselection in contrast to merely static preselection.

Keywords

Multi-objective optimization, fitness model, surrogate model, model assisted, surrogate assisted, data based model, evolutionary algorithm, evolution strategies.

1 Introduction

Evolutionary algorithms are able to find good solutions for complex and high dimensional optimization problems, but require a substantial number of evaluations of the fitness function due to the partial random exploration of the optimization space. In evolutionary hardware-in-the-loop optimization of control systems, such as Krettek et al. (2007), a single fitness evaluation takes several seconds. Obviously, extensive evaluations, simulations, or calculations of the fitness function constrain the feasible population size and number of generations in the evolutionary optimization of technical systems.

The literature contains many proposals for fitness models to partially substitute true evaluations of the fitness function. A significant speed up is achieved for problems for which the fitness estimation based on the model is substantially faster than the true evaluation of the process itself, under the condition of a fitness landscape that is structured enough to be modeled on the base of a limited dataset. The efficient global optimization (EGO), originally proposed by Jones et al. (1998), first distributes fitness evaluations uniformly across the search space. From these uniform samples it builds a global model and evaluates the most promising solution according to the model. This cycle of single best expected fitness evaluation and model refinement effectively reduces the number of required fitness evaluations. The original scalar approach has been extended to multi-objective optimization problems by Knowles (2006), Keane (2006), and Ponweiser et al. (2008). The first two approaches generate separate models for the different objectives and select best candidates on the basis of weighted aggregation of objective estimates. In the third approach, selection is based on the estimated improvement of the S-metric (Zitzler and Thiele, 1998).

Estimation of distribution algorithms also employ fitness models (Larranaga and Lozano, 2001; Zhang et al., 2008). Being related to model assisted evolutionary algorithms, they abandon the conventional mutation and recombination operators and instead continuously adapt a probability distribution from which they sample new candidate solutions. The parameters of this distribution are updated on the basis of statistic information obtained from previous evaluations and selection decisions. In that way, the posterior distribution implicitly captures knowledge about the underlying fitness function and drawing from this distribution generates offspring that are superior to mere blind random variations of the parents.

In contrast, model assisted evolutionary algorithms employ blind mutation and recombination but utilize a data based model to preselect promising candidates afterward. In summary, estimation of distribution algorithms use the model for systematic and informed sampling of new variants, whereas model assisted algorithms first sample blindly and then use the model to discard supposedly inferior solutions prior to the actual evaluation. Since the density of fitness samples is sparse in a high-dimensional optimization space, it can be futile to learn a global fitness model from relatively few samples. Fitness models might induce misleading local minima that the original fitness function does not possess. In order to avoid premature convergence into deceptive minima, it is mandatory to continuously verify model quality based on true evaluations (Jin et al., 2002). Model management decides which solutions are discarded in advance due to their inferior estimated fitness and which remaining promising candidates are subject to a true evaluation. If the model quality is high, selection mostly relies on the fitness model; if the quality decreases the number of true fitness evaluations is increased. As these additional fitness evaluations provide further training data to refine the data based model, the model quality is expected to reimprove in subsequent generations.

There are two basic concepts of model management presented in Jin (2005) and Jin et al. (2002). The generation based approach evaluates the true fitness function over the course of some *controlled generations* and then relies on the model for the next couple of generations. The model is trained based on the examples provided by the true evaluations generated during the *controlled generations*. The frequency of invoking controlled selection depends on the model quality. The period of *uncontrolled selection* is adjusted to model quality in order to avoid convergence into deceptive local minima of the model. In contrast to the generation based approach, the individual based approach evaluates a certain fraction of the current population according to the true fitness function and relies on the estimated fitness for the remaining ones.

The comparative analysis of individual based strategies in Gräning et al. (2005) favors preselection (or prescreening) as the most robust and efficient scheme. For this reason, preselection is employed by the majority of approaches, such as Zhou et al. (2005), which employs a hierarchical scheme that combines a global with a local model. Ulmer et al. (2004) present a strategy for single objective optimization, controlling preselection pressure by the number of prescreened individuals. Their original approach was later improved by Hoffmann and Hölemann (2006), adapting the number of real evaluations rather than the pre-offspring size. Model assisted evolution has been applied to multi-objective optimization in Emmerich and Naujoks (2004) and Emmerich et al. (2006). Our previous approaches controlling the preselection pressure for scalar problems in Hoffmann and Hölemann (2006) is adapted to the novel multi-objective scheme presented here.

In single objective problems, the relationship between the model and the selection error is straightforward. The model quality is defined in terms of the difference in the two rankings based according to estimated and true fitness, as stated in Jin et al. (2003). In the scalar case, λ-control compares the observed ranking errors with those of a purely random selection and adjusts the preselection pressure accordingly.

In multi-objective optimization, fitness based ranking becomes obsolete and selection rather relies on the concept of dominance. The influence of model errors in the multiple objectives on the imposed dominance relations among solutions is considerably more complex and thus demands an appropriate redefinition of the model quality for multi-objective optimization.

Sections 2 and 3 detail the principal operation of an evolutionary algorithm with model assisted preselection. Benchmark functions are introduced in Section 4. The effect of alternative data based models on model accuracy and hence preselection quality is analyzed in Section 5, which also introduces model quality for multi-objective problems. Methods for the preselection of individuals in multi-objective evolutionary optimization are given in Section 6. The efficiency of model management is significantly increased by the novel type of preselection, the so called λ-control, proposed in this paper. Section 7 extends the concept of model quality control to multi-objective problems, and analyzes the efficiency and limitations of λ-control in terms of reduction of true fitness evaluations.

2 Multi-Objective Evolutionary Optimization

Real world optimization problems often encompass multiple conflicting objectives. A solution dominates another solution if it is superior in at least one objective and not inferior in any other objective. In this case, improvement of a Pareto optimal solution in one objective comes at the cost of degradation in at least one other conflicting objective. A multi-objective evolutionary optimization algorithm (MOEA) shown in Figure 1 generates a diverse and representative set of approximately optimal compromise solutions,

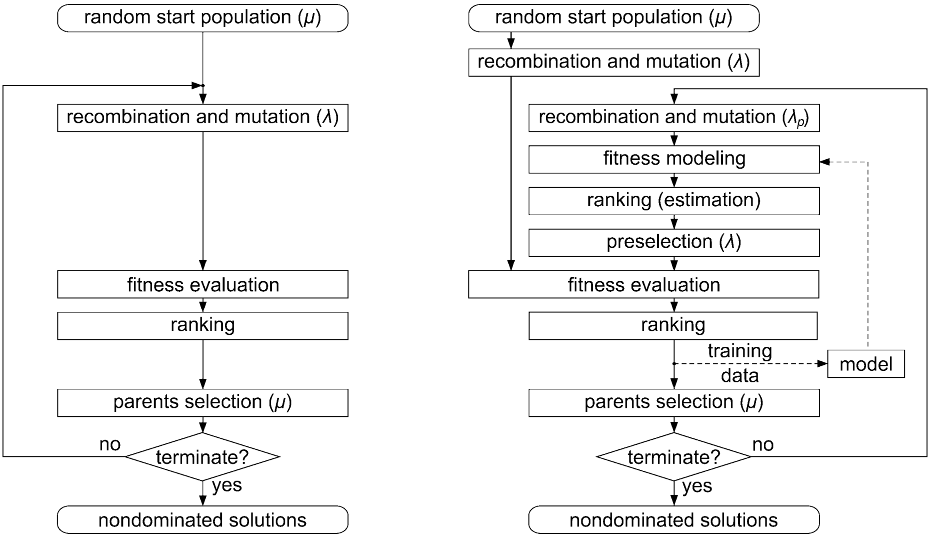

Figure 1: Evolutionary loop of standard ES (left) and MAES (right).

the so called nondominated solutions. The expert selects the solution that he or she considers best in retrospect according to preference. Mutation and recombination in the multi-objective case are similar to the scalar evolutionary optimization; the main difference between both problem types is the selection.

Fitness model assistance is applicable to any evolutionary optimization scheme including multi-objective approaches such as SPEA 2 (Zitzler et al., 2001) or NSGA-II (Deb et al., 2002). Our MOEA framework, later denoted by *standard ES*, is derived from the SPEA 2 and NSGA-II schemes. It employs an evolutionary strategy with elitism, discrete parameter recombination, and self adaptive mutation step sizes originally proposed by Schwefel (1981). In recombination, single parameters are drawn at random from either of the two parents. In the original scalar evolution strategy, solutions are ranked according to fitness and the μ best solutions become parents. In multi-objective problems, such a ranking is not possible, as the criteria only allow a partial ordering. Therefore, the ranking based (μ,λ)-selection is replaced by a dominance based scheme highly similar to NSGA-II. Selection prefers individuals who are not dominated by any other member of the population. The rank of an individual i is defined as $r_i = 1 + n_i$ in which n_i denotes the number of solutions dominating the ith individual. All nondominated individuals obtain rank one and therefore the best fitness. In order to achieve a good approximation of the true Pareto front, the nondominated solutions should spread uniformly across the Pareto front. Therefore, selection penalizes similar solutions in order to maintain a diverse population. At first, all nondominated solutions of rank one are selected as part of the intermediary set. If the size of the set exceeds the number of parents μ, then the set is reduced in an incremental manner by identifying the closest pair of solutions, and removing the solution with the nearest neighbor. This is repeated until the μ most dissimilar solutions remain, while individuals on the edges of the Pareto front are always kept. Nondominated solutions are stored in an elite archive. The size of the archive is limited using the same mechanism just described. It constitutes the final solution set after optimization. Similar to the classic $\mu + \lambda$ strategy, the archive constitutes the pool

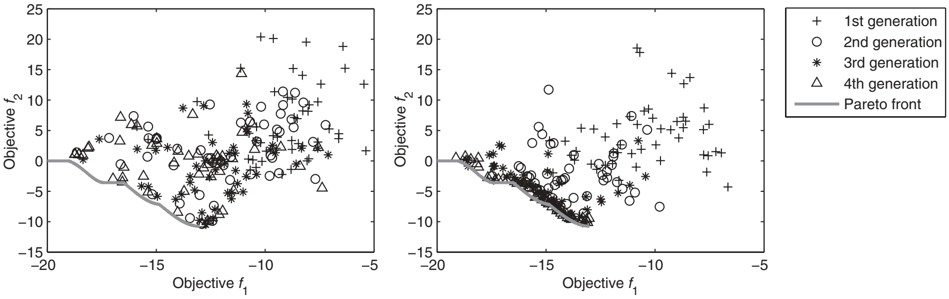

Figure 2: Evolution of the population for a standard evolution strategy (left) and a model assisted multi-objective evolution strategy (right) over four generations. Benchmark function: Kursawe with $n = 3$ (Kursawe, 1991).

from which parents are selected. This elitism strategy offers the advantage that once achieved, a population spread over the Pareto front is maintained in future generations. This property is of particular advantage for the λ-control presented in Section 7.

Even though our evolutionary optimization scheme is slightly different from the SPEA 2 and NSGA-II framework, the performance gain attributed to controlled model assistance should be similar. Those schemes that utilize past fitness evaluations for a more informed generation of new variants such as CMA-ES from Hansen and Ostermeier (2001), are expected to benefit less from model assistance as the distribution from which new variants are sampled implicitly contains prior knowledge on the local fitness landscape. A general survey of multi-objective evolutionary optimization methods is provided by Zitzler and Thiele (1998), Deb (2001), and Coello (2006). A standard evolutionary strategy is extended to a model assisted evolutionary strategy (MAES) by introducing a preselection step based on estimated fitness prior to the true fitness based selection. The principal operation is illustrated in the right part of Figure 1. In contrast to the standard evolutionary strategy, $\lambda_p > \lambda$ rather than only λ offspring are generated via recombination and mutation of parents. Their fitness is estimated by the data based model. Only the λ solutions which the model is not able to discriminate undergo a true fitness evaluation. At the same time, the new evaluations provide additional training samples to refine the fitness model online.

3 Model Assisted Multi-Objective Evolutionary Optimization with Preselection

Compared to the usual λ individuals, a larger number of offspring λ_p is created, increasing the chance of a model assisted strategy to discover superior solutions. Under the assumption of perfect fitness estimation, the model assisted strategy converges at a rate that is equivalent to a standard evolution strategy with λ_p offspring only requiring a fraction λ/λ_p of true fitness evaluations. The analysis of the model assisted scheme on benchmark problems presented in the following confirms that with proper model management, a substantial portion of the theoretical feasible gain is actually realized. Figure 2 shows the evolution of the population in the objective space for the first four generations of an evolutionary run. The two graphs compare the evolution of a standard (10 + 50)-evolution strategy (left) with a (10+50)-model assisted evolution strategy (right) with $\lambda_p = 200$ preoffspring individuals. Note that both schemes require the same computational effort in terms of number of true fitness evaluations. It is apparent that the model assisted variant converges much faster to the Pareto front.

4 Benchmark Problems

In order to analyze the convergence properties of alternative evolutionary strategies, several runs are performed on a set of multi-objective benchmark minimization problems well known in the literature: Kursawe (KUR) in Kursawe (1991) with $-4 \leq x_i \leq 4$ for $i = 1, 2, \ldots, n$ and $n = 6$, FON in Poloni et al. (2000) with $-4 \leq x_i \leq 4$ for $i = 1, 2, \ldots, n$ and $n = 7$, and FES3 in Fieldsend et al. (2003) with $0 \leq x_i \leq 1$ for $i = 1, 2, \ldots, n$ and $n = 5$. If not stated otherwise, the used MOEA operates with $\lambda_p = 400$, $\lambda = 100$, and $\mu = 20$, which is suitable for conventional and model assisted strategies. The parents are selected from an elite archive limited to 100 representative solutions. The default method used for preselection is *simulated elite membership* (see Section 6). The fitness is estimated by a locally weighted linear regression model *LWR 5n* that is progressively refined online as new fitness evaluations become available (see Section 5.1). The reported results are based on the average of 50 independent optimization runs.

The convergence of the population toward the Pareto front is measured by the so called *S-metric* in Zitzler and Thiele (1998). The S-metric is defined as the (hyper-) volume of the optimization space that is dominated by a set of individuals. The following comparisons are with respect to the residual S-metric defined as the difference between the ideal S-metric of the Pareto front and the actual S-metric covered by the elite population. The goal of multi-objective optimization is to reduce the area between the theoretical Pareto front and the area dominated elite population. In the case of λ-control, the number of true fitness evaluations λ varies across generations. For a fair comparison that captures the actual computational effort, the evolution of the residual S-metric is always plotted with respect to the number of true fitness evaluations rather than generations.

5 Fitness Modeling

Fitness modeling estimates the m unknown true objectives $f_j(\mathbf{x}_q)$, $j = 1, 2, \ldots, m$ of a solution $\mathbf{x}_q$ in a search space of dimension n, derived from the known objective values of k training datasets $(\mathbf{x}^{(i)}, f_j(\mathbf{x}^{(i)}))$, $i = 1, 2, \ldots, k$ stored in the database D. Instance based learning methods delay the explicit generation of a model until the time of query and instead only store the incoming training data. Upon a new query, a local model is generated from those training items that are most similar to the query point. To compensate for different ranges of value, the range of input data is normalized to a unit interval $[0, 1]$ across all dimensions. For all following considerations, this normalized space is used.

Fitness models for multi-objective problems are identical to their scalar counterpart, with the sole difference that a separate fitness model is generated for each objective. Note that the dominance relationship between solutions is affected by all objectives simultaneously. A single incorrect estimated objective might turn a dominated solution into a nondominated one if the estimate in this objective is overly optimistic. Obviously, the ability of the fitness model to correctly identify the set of nondominated solutions in the current population depends on the quality of the worst objective model.

5.1 Local Linear Weighted Regression

To estimate the objective value of a new query point $\mathbf{x}_q \notin D$, the *local weighted regression* (LWR) generates a local linear model, from those training items in D which in input space are most similar to the query $\mathbf{x}_q$ (Atkeson et al., 1997). The unknown function $f(\mathbf{x}_q)$ is locally approximated by linear regresson in a neighborhood of the query point

$$\hat{f}(\mathbf{x}_q, \boldsymbol{\beta}) = \beta_0 + \sum_{l=1}^{n} \beta_l x_l. \tag{1}$$

The coefficients $\boldsymbol{\beta}$ of the linear regressor $\mathbf{x}_l$ are selected such that the weighted quadratic error E is minimized over the set of training data

$$E = \sum_{i=1}^{k} w_i\left(\mathbf{x}_q, \mathbf{x}^{(i)}\right)\left(f\left(\mathbf{x}^{(i)}\right) - \hat{f}\left(\mathbf{x}^{(i)}, \boldsymbol{\beta}\right)\right)^2 \to \min. \tag{2}$$

The error of individual training items is scaled by the weight $w_i(\mathbf{x}_q, \mathbf{x}^{(i)})$ according to the similarity of the query point $\mathbf{x}_q$ with the data item $\mathbf{x}^{(i)}$. The solution that minimizes the weighted quadratic error is given by

$$\boldsymbol{\beta} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{\text{-1}} \mathbf{H}^T\mathbf{W}\mathbf{y}. \tag{3}$$

The matrix $\mathbf{H}$ depends only on the data points $\mathbf{x}^{(i)}$ and the corresponding fitness values $\mathbf{y}$.

$$\mathbf{H} = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(k)} & x_2^{(k)} & \cdots & x_n^{(k)} \end{bmatrix}, \quad \mathbf{y} = \left[f\left(\mathbf{x}^{(1)}\right), f\left(\mathbf{x}^{(2)}\right), \ldots, f\left(\mathbf{x}^{(k)}\right)\right]^{\mathrm{T}} \tag{4}$$

The linear approximation model obtains its local characteristics from the weight matrix $\mathbf{W} = \text{diag}[w_1, w_2, \ldots, w_k]$ in Equation (3). The weight w_i increases with the similarity of the data point $\mathbf{x}^{(i)}$ and query point $\mathbf{x}_q$ determined in terms of their Euclidean distance

$$w_i\left(\mathbf{x}_q, \mathbf{x}^{(i)}\right) = \exp\left(-\left(\frac{||\mathbf{x}^{(i)} - \mathbf{x}_q||_2}{\sigma(\mathbf{x}_q)}\right)^2\right). \tag{5}$$

The density of training data increases as with progressive evolution more fitness training points are added to the model. The local density of training items in the current population increases further with convergence of the population to the Pareto front. The increase in density is compensated for by a dynamic scaling of the weighting function such that the neighborhood radius

$$\sigma(\mathbf{x}_q) = \min_{i=1,2,\ldots,k}\left(||\mathbf{x}^{(i)} - \mathbf{x}_q||_2\right) \tag{6}$$

is adjusted to a distance between the query point and its nearest training item. Due to the dynamic scaling, the training item most similar to the query always obtains the same weight of $w_i = \mathrm{e}^{-1}$. Dynamic scaling guarantees that the local model is fitted to roughly the same number of nearest training points irrespective of their actual density. All incoming fitness evaluations are stored as training data online. As the scaling factor decreases over time, fewer training items actually contribute to the model. In order to limit the computational burden of inverting the data matrix **H**, the local model is only built from a subset of nearest neighbors. The size of the subset in this paper is chosen as either $2n$ or $5n$. The use of $5n$ items is almost equivalent to the use of the full database, hence distant points contribute marginally because of the weighting, while $2n$ requires considerably less computational afford.

5.2 Alternative Data Based Regression Models

A distance weighted average employs a constant rather than a linear regression function

$$\hat{f}(\mathbf{x}_q) = \frac{\sum_{i=1}^{k} f\left(\mathbf{x}^{(i)}\right) \cdot w_i\left(\mathbf{x}_q, \mathbf{x}^{(i)}\right)}{\sum_{j=1}^{k} w_j\left(\mathbf{x}_q, \mathbf{x}^{(j)}\right)}. \tag{7}$$

The model is simpler and requires fewer data points. On the other hand, a distance weighted average is not capable of extrapolation beyond the region populated by the samples. The nearest neighbor method generates no model at all but merely takes the training data that are closest to the query point as the prediction. Conceptually, the nearest neighbor model corresponds to the distance weighted average in which the closest training data obtain a weight of one and all other weights become zero.

Gaussian Processes or *Kriging* have been proposed by Ulmer et al. (2003) for fitness modeling. They offer the advantage not only to predict the fitness, but also to estimate the associated uncertainty of the prediction. This information allows it to estimate the *probability of improvement* with respect to the current best fitness and use this likelihood rather than absolute fitness in the preselection step, which is the method used in almost all recent studies in this field such as Emmerich and Naujoks (2004) and Ulmer et al. (2004). Unfortunately, the computational demand to compute a Gaussian Process model is fairly large, as it requires a nonlinear optimization to identify its parameters. The analysis of model quality in Hoffmann and Hölemann (2006) shows that Gaussian Processes do not offer a significant advantage compared to local linear weighted regression models. Nevertheless, it remains an open issue as to whether preselection methods that consider probabilities of improvement in fitness are superior to those that only rely on the fitness estimate itself. From our experience, the actual selection decisions of both approaches are highly similar once there is a sufficient amount of training data.

5.3 Model Quality

An important aspect of model management is the monitoring of the model quality. In the case of fitness modeling, the absolute model error is less important compared to the ability of the model to take correct selection decisions. It is thus necessary to analyze the impact of the model error on the selection error.

In the multi-objective case, the individual criteria model errors affect the correct prediction of the dominance relations between pairs of individuals. Two solutions A

and B assume either one of three possible dominance relationships: A dominates B ($A \succ B$), B dominates A ($B \succ A$), or neither dominates the other.

Now let us consider the estimated objective values of A and B, the predicted dominance relation based on these estimates, and compare it with the true dominance relationship based on the true objective values, using the function d, where $d(A, B) = 1$ if A dominates $\mathbf{f}_j$, and zero otherwise. From the error in the predicted objective value, it is possible to calculate the relative frequency of an incorrectly predicted dominance relationship. The estimated dominance relation $d(\hat{A}, \hat{B})$ is determined from the model and compared with the true dominance $d(A, B)$ according to the true objective values. The frequency of incorrect dominance predictions across the entire population is given by:

$$P(d(A, B) \neq d(\hat{A}, \hat{B})) = \frac{\sum_{i=1}^{\lambda} \sum_{j=1}^{\lambda} \mathrm{abs}(d(\mathbf{f}_i, \mathbf{f}_j) - d(\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j))}{\lambda(\lambda - 1)}. \tag{8}$$

Even a model that takes completely random predictions nevertheless often coincidentally gets the dominance relationship right. Actually, for a large number of criteria, almost no individual is dominated or dominates another, and the random model will get this *obvious* relationship right with high probability. Therefore, the above model quality is normalized with the probability of incorrect predictions of a random model. The probability of a random solution to dominate another random solution in a single objective is $P = 1/2$. Accordingly, the probability of a random solution dominating another random one in all m objectives is:

$$P_Z(C \succ D) = \left(\frac{1}{2}\right)^m. \tag{9}$$

In high dimensional objective spaces, solutions rarely dominate each other. In contrast, the frequency of a dominance in the actual nonrandom population is given by:

$$P_R(A \succ B) = \frac{\sum_{i=1}^{\lambda} \sum_{j=1}^{\lambda} d(\mathbf{f}_i, \mathbf{f}_j)}{\lambda(\lambda - 1)}. \tag{10}$$

When a model with random output is assumed, incorrect dominance predictions occur in the case of a real domination while there is none in the random data, which has the probability of $P_R(1 - P_Z)$, and in the case where there is no real dominance while random solutions dominate each other, occurring in the probability $(1 - P_R)P_Z$. Therefore, the expected frequency of an incorrect dominance prediction that stems from random objective values rather than their true counterparts is given by:

$$P(d(A, B) \neq d(C, D)) = P_R(1 - P_Z) + (1 - P_R)P_Z. \tag{11}$$

The model quality at generation g is defined as the difference between the frequency of incorrect predictions of a random model and the same frequency of the actual model, normalized by the frequency of a random model

$$Q^{(g)} = \frac{P(d(A, B) \neq d(C, D)) - P(d(A, B) \neq d(\hat{A}, \hat{B}))}{P(d(A, B) \neq d(C, D))}, \tag{12}$$

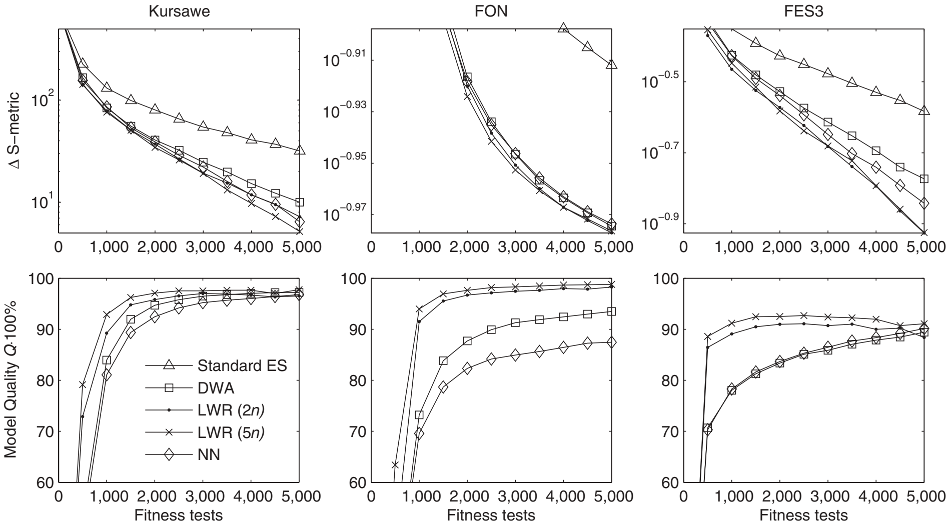

Figure 3: (Top) Evolution of the residual S-metric of a standard ES and a MAES for regression models: nearest neighbor (NN), distance weighted average (DWA), and linear weighted regression (LWR). The LWR uses either $2n$ or $5n$ nearest neighbors to build the local model. (Bottom) Model quality Q in percent.

such that $Q^{(g)} = 1$ becomes one in case the fitness predicts all dominance relationships correctly. A quality of $Q^{(g)} = 0$ indicates that the current fitness model does not outperform merely random predictions, while negative values indicate that the use of the model is inferior to selecting λ individuals out of the preoffspring randomly.

Computing the frequency of correctly predicted dominance relationships among all solutions requires a true fitness evaluation of each member of the current population. However, the true objective values are only known in retrospect for the subset of $\lambda < \lambda_p$ solutions that passed the preselection. Therefore, the model quality of the global population is approximated by the model quality of this subset. It is assumed that this subsampling of model quality is still an appropriate indicator of global model quality.

5.4 Comparison of Fitness Models

This section analyzes the impact of the type of regression model on the convergence property of model assisted multi-objective evolution strategies. Figure 3 shows benchmark results that compare a standard evolution strategy and model assisted variants with distance weighted, nearest neighbor, and linear weighted regression with $2n$ or $5n$ nearest training points. The upper row shows the evolution of the residual S-metric for the three benchmark problems presented in Section 4, and the lower row shows the evolution of model quality.

The analysis indicates that all fitness models result in a significant improvement over the standard evolution strategy. For the same number of true evaluations, the residual S-metric is much lower, as shown by the logarithmic scale in Figure 3, or vice versa the same residual S-metric is achieved with a much smaller number of evaluations. The model assisted variants achieve the same S-metric within approximately 2,000 evaluations compared to 5,000 evaluations of the standard evolution strategy. Analyzing the different regression models in detail, the linear weighted regression is favored for model assisted optimization as it exhibits superior model quality, which results in faster convergence toward the Pareto front. The model quality increases significantly during the first 10 generations, after which the models are based upon approximately 1,000 data points. In subsequent generations, model quality only improves gradually, due to the fact that as solutions become more similar, the relative order of the objective values is harder to predict. Eventually, model quality saturates at a level of about 90–95%. The connection between model quality and faster convergence of the evolutionary algorithm is apparent. Fast convergence of the optimization goes along with high model quality. This observation confirms that our definition of model quality is appropriate for model management in the context of model assisted multi-objective evolutionary optimization. For the used benchmark functions, the local linear regression model outperforms the distance weighted average and nearest neighbor model due to its ability to extrapolate. Extrapolation is crucial for fitness modeling of continuous functions, because the offspring solutions are often located at or beyond the boundary of the training data generated in previous generations. Global extrapolation is problematic in high dimensional spaces and for nonlinear functions with a limited set of training data. In the context of fitness modeling, the global approximation plays a minor role, as the query and training points are usually confined to a small region in the vicinity of the optimum instead of a uniform distribution across the entire domain. Furthermore, selection and dominance are based on the relative ranking of objectives rather than the absolute objective values. Thus, an absolute but constant model error does not affect the dominance relationship.

6 Preselection Methods

In principle, the common selection methods for multi-objective evolution strategies are equally suitable for the preselection of λ best candidate solutions from the offspring population of size λ_p. Preselection not only affects the future convergence of optimization, but also the distribution of novel training data. This observation results in two requirements on preselection.

1. Selection of solutions that are not dominated by the current elite, to achieve a rapid convergence toward the Pareto front.
2. Selection of dissimilar solutions that differ from the current training dataset and the current population. These novel solutions contain valuable information to improve the quality of the model. Multiple copies of similar solutions do not improve the model accuracy, and are thus wasted fitness evaluations in terms of information gain.

The following proposed preselection methods emphasize these two aspects of dominance and novelty in slightly different manners.

- **Standard Selection.** Solutions are selected in the same manner as the regular parents, namely, preferring solutions that are least dominated by others. In the case of an excess of nondominated solutions, incremental thinning of nearest neighbors is applied to reduce the set.

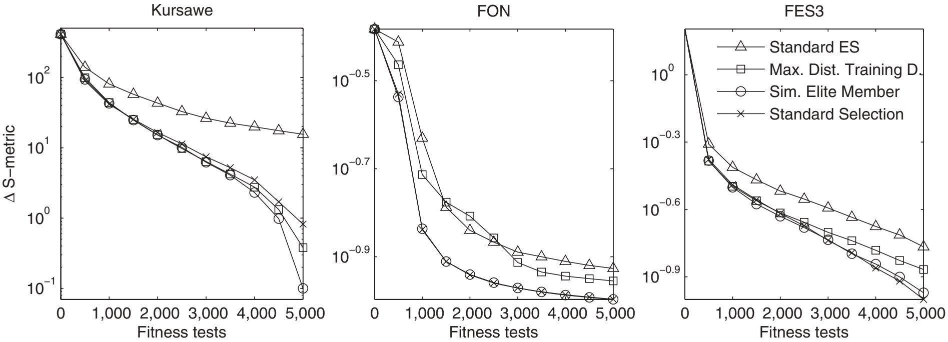

Figure 4: MAES benchmark runs for different preselection methods.

- **Simulated Elite Membership.** Dominance ranking is applied to the union of the elite set with known fitness and the λ_p preoffspring individuals with their estimated fitness. Again, those solutions are selected that are either nondominated or least dominated by the elite or members of the current population (see Section 2). A solution, even though it is not dominated within its own population, might still be dominated by members of the elite archive. Such a solution does not contribute to the improvement of the residual S-metric. A preselection scheme that immediately considers the elite archive is more likely to generate solutions that eventually are not dominated by the elite and thus improve the S-metric.

- **Maximal Distance to Training Data.** The degree of dominance of the λ_p preoffspring solutions is determined with respect to the elite archive only and the nondominated solutions are selected. In case the nondominated set is larger than λ, the solutions with the largest minimal distance to the training dataset are selected. This scheme prefers the most novel variants among all nondominated solutions.

The benchmark results in Figure 4 illustrate similar good performance results of the *standard selection* and the *simulated elite membership* method. The performance of the *maximal distance to training data* method is inferior, as a substantial number of true fitness evaluations are wasted on yet unexplored but also inferior regions of the search space. The simulated elite membership method demonstrates a robust performance irrespective of the problem type and is thus employed in the following analysis.

7 Controlled Model Assisted Evolution Strategies

First let us analyze the influence of the selection pressure in the preselection and actual selection on the convergence of the model assisted evolution strategy. Figure 5 shows several benchmark runs for different sizes of the offspring population. Note that this size effectively regulates the selection pressure in preselection, as more offspring means more competition for the λ_p spots. Apparently, the convergence is accelerated with more offspring. This means that the models are able to distinguish between similar solutions, as otherwise the model would no longer be able to discriminate similar offspring solutions. The selection pressure in preselection also depends on λ, the number of

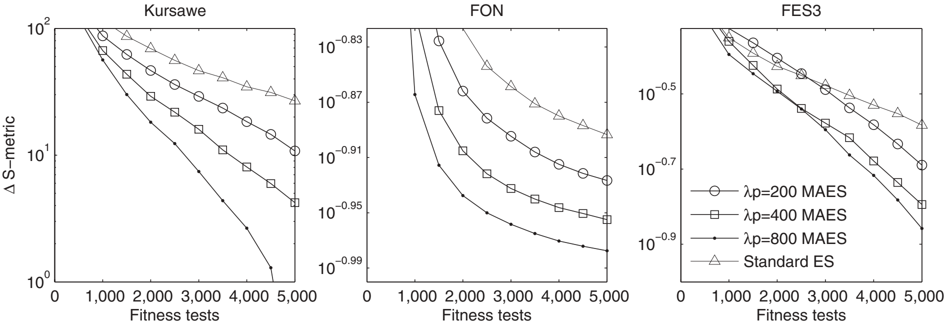

Figure 5: Benchmark results of MAES with $\lambda_p = 200, 400, 800$ and a standard ES.

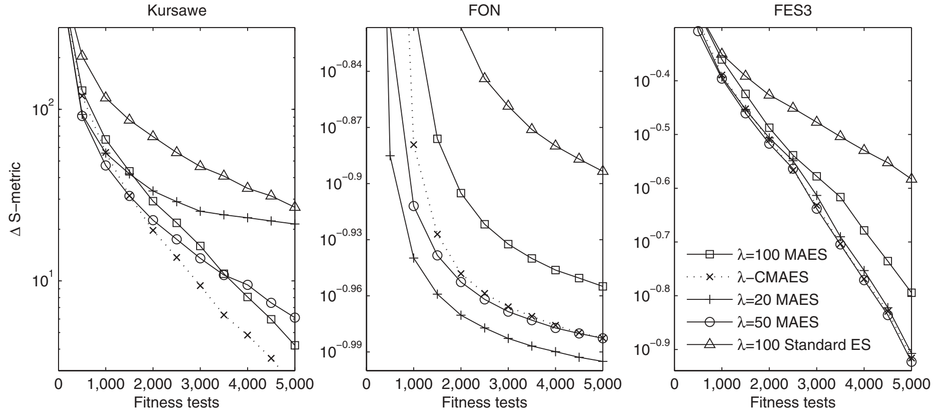

Figure 6: Benchmark comparing the λ-cMAES with different uncontrolled MAES and a standard ES.

solutions that survive preselection. Benchmarks for different values of λ are illustrated in Figure 6. The parameter λ has a significant influence on the convergence, which turns out to be problem specific. Thus, there is no general parameter that is optimal across all problem types. The strategy with $\lambda = \mu = 20$ best illustrates this phenomenon. This variant is superior to the $\lambda = 50$ and $\lambda = 100$ settings on the FON problem, but exhibits premature convergence on the Kursawe function. The significant variation in convergence is attributed to the different structure of the optimization problems. In the case of the Kursawe problem, the Pareto front is composed of several disconnected segments that belong to isolated regions in the parameter space. To approximate the Pareto front on global scale, an exhaustive search across the entire parameter space is mandatory, thus the evolutionary algorithm should focus on exploration in the initial stages. A small intermediate population size results in the loss of the entire set of representative solutions in some subregions that are not recovered in later generations. The FON problem possesses a connected Pareto front, such that convergence toward the front is more crucial in the early stages than maintaining diversity. The efficient exploitation of promising solutions is supported by a small intermediate population size λ, which

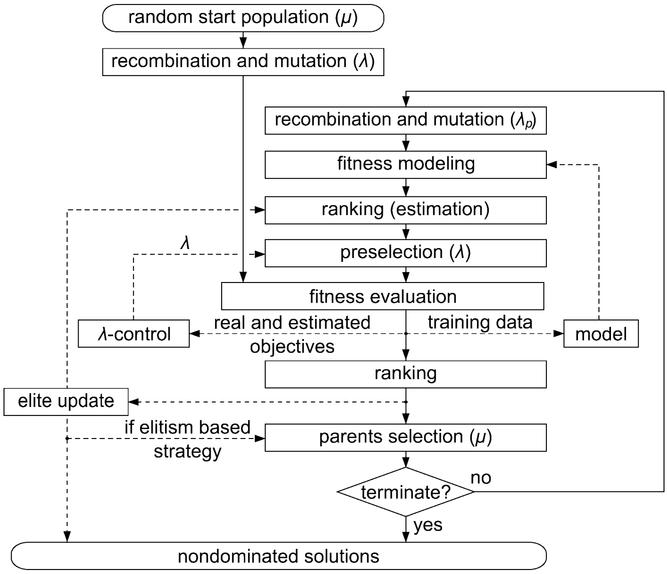
random start population (μ)
recombination and mutation (λ)
recombination and mutation (λ_p)
fitness modeling
ranking (estimation)
λ
preselection (λ)
fitness evaluation
λ-control
real and estimated objectives
training data
model
ranking
elite update
if elitism based strategy
parents selection (μ)
terminate?
no
yes
nondominated solutions

Figure 7: Evolutionary loop of the λ-controlled model assisted multi-objective evolutionary strategy.

more effectively utilizes true fitness evaluations per generation. As exploitation is more important than exploration, the search should rely more strongly on the model. The results in Figure 6 show that the optimal selection pressure in preselection is problem specific, as it regulates the amount of exploration versus exploitation in search. The proper selection of λ requires metaknowledge about the underlying structure of the optimization problem, which is usually not available in advance. The novel idea proposed in this paper is to adjust the parameter λ and thereby the selection pressure during evolution. The proposed λ-controlled model assisted evolution strategy (λ-cMAES) adjusts the number of true fitness evaluations to the current model quality (Hoffmann and Hölemann, 2006). The MOEA algorithm with integrated λ-control is depicted in Figure 7. It is basically an extension of the model assisted algorithm with a feedback of the model quality. In principle, the number of true evaluation might vary between the size of the offspring population λ_p, in which case the model is basically ignored; and the number of parents μ, in which case effectively all selection decisions are taken by the model. The basic idea is to rely more on the model if its quality is high and more on true evaluations if the model quality is low. The fitness model is able to at least partially estimate the correct dominance relations if the observed ratio of the incorrect decision is lower than the expected error of a random model.

Therefore, the number of true evaluations $\lambda^{(g+1)}$ is reduced if the model based selection decisions are superior to a random decision, while λ is increased if the model is inferior to random selection

$$\tilde{\lambda}^{(g+1)} = \lambda^{(g)} \left(\frac{1}{2}\right)^{Q^{(g)}\delta_\lambda}. \qquad (13)$$

The gain δ_λ determines the rate of adaptation and is chosen as $\delta_\lambda = 0.5$, allowing the algorithm to adapt in a few generations, while avoiding a too strong oscillation of

λ. Zero model quality is equivalent to a random model, thus positive $Q^{(g)}$ reduces λ, whereas negative $Q^{(g)}$ increases λ. The parameter λ cannot exceed the number of offspring λ_p and cannot fall short of the number of parents μ.

$$\lambda^{(g+1)} = \begin{cases} \lambda_p & \text{for } \tilde{\lambda}^{(g+1)} > \lambda_p \\ \mu & \text{for } \tilde{\lambda}^{(g+1)} < \mu \, . \\ \tilde{\lambda}^{(g+1)} & \text{else} \end{cases} \tag{14}$$

Figure 6 illustrates the superior performance of the λ-control variant over its counterparts with constant λ. The self adaptive scheme with the λ-controlled MAES demonstrates a robust good performance across the three optimization problems. Depending on the particular problem, it saves up to 30 to 70% of true fitness evaluations compared to a conventional evolution strategy. It is superior over all its static counterparts in the case of the Kursawe and FES3 problem. This superiority is attributed to the fact that the optimal parameter setting for λ varies with the state of convergence during an evolutionary run, shifting from exploration in earlier stages to exploitation in later generations. The dependency of the parameter λ on the problem type and the progress of optimization is confirmed in the benchmark experiments. For the FON problem, the setting with constant minimal $\lambda = 20$ performs better than the controlled variant. The FON problem is the least complex, such that model quality is always high, and thus true evaluations should be kept to an absolute minimum.

Almost all experiments that rely on the λ-controlled model assisted evolution strategies with a comma strategy run into the problem of premature convergence to subregions of the Pareto front. The first few generations evolve properly, upon which with improved model quality the population size λ is reduced. In this early stage, some regions of the Pareto front appear more promising than others, due to the small amount of training data. In a kind of self-fulfilling prophecy, solutions are selected from regions already more advanced toward the Pareto front, which in turn favors the local progress of these very solutions. As a result, the comma strategy pools solutions in these coincidental *winning* regions, which leads to reduced diversity and premature convergence. For this reason, the comma strategy is not advisable in conjunction with the λ-cMAES. In the multi-objective case, plus and elitism strategies maintain the diversity of solutions and the search uniformly explores the Pareto front.

8 Summary

This paper presented a novel scheme for controlled model assisted evolutionary optimization, which extends previous approaches from the scalar case to multi-objective scenarios. The main contribution is a proper definition of model quality for multi-objective problems in terms of correctness of predicted dominance relationships. This definition provides the basis to enable model management to adapt the selection pressure in model based preselection to the model quality. The number of true fitness evaluations is significantly reduced by delegating a substantial proportion of decisions to the fitness model. The regulated balance between estimated and true fitness based selection, the so called λ-control, achieves an optimal utilization of precious fitness evaluations. The advanced model management finds the problem specific optimal balance between exploitation of the fitness model and exploration based on true fitness evaluations. Only those solutions that the model is not able to discriminate are tested on the true fitness function. The convergence and robustness of the λ-cMAES strategy is advantageous compared not only to a standard evolution strategy but to the model assisted scheme with static, uncontrolled preselection as well. The feasibility and effectiveness of the method are evaluated on benchmark problems, which indicate a reduction of true fitness evaluations to about 30–70% of the original effort. In the future, we intend to augment the two stage selection to a three stage process in the context of hardware-in-the-loop optimization of technical systems. The first stage is constituted by preselection with a data based fitness model as described in this paper. The preselected solutions undergo a simulation as a surrogate test, which is computationally less demanding than a hardware-in-the-loop evaluation. Only the remaining best candidates are eventually evaluated using hardware in the loop. The information gained in the hardware-in-the-loop trials is utilized to refine the data based fitness model and the simulation as well. This three stage approach promises to open an avenue toward efficient evolutionary optimization and design of technical systems. The conventional engineering development cycle consists of specification, design, and implementation. The ultimate goal is to support the engineer in the second design step, by delegating the exploration of the design space to an evolutionary algorithm. The designer focuses on decision making by comparing evolutionary generated alternative designs rather than searching for them.

References

Atkeson, C., Moore, A., and Schaal, S. (1997). Locally weighted learning. *AI Review*, 11:11–73.

Coello, C. C. (2006). Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36.

Deb, K. (2001). *Multi-objective optimisation using evolutionary algorithms*. Chichester, UK: John Wiley.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197.

Emmerich, M., Giannakoglou, K., and Naujoks, B. (2006). Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439.

Emmerich, M., and Naujoks, B. (2004). *Metamodel-assisted multiobjective optimisation strategies and their application in airfoil design* (pp. 249–260). Berlin: Springer.

Fieldsend, J. E., Everson, R. M., and Singh, S. (2003). Using unconstrained elite archives for multi-objective optimisation. *IEEE Transactions on Evolutionary Computation*, 7:305–323.

Gräning, L., Jin, Y., and Sendhoff, B. (2005). Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study. In *Proceedings of the 13th European Symposium on Artificial Neural Networks (ESANN 2005)*, pp. 273–278.

Hansen, N., and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.

Hoffmann, F., and Hölemann, S. (2006). Controlled model assisted evolution strategy with adaptive preselection. In *Proceedings of the IEEE International Symposium on Evolving Fuzzy Systems*, pp. 182–187.

Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12.

Jin, Y., Hüsken, M., and Sendhoff, B. (2003). Quality measures for approximate models in evolutionary computation. In *Proceedings of the Birds of a Feather Workshop, Genetic and Evolutionary Computation Conference (GECCO 2003)*, pp. 170–173.

Jin, Y., Olhofer, M., and Sendhoff, B. (2002). A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494.

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

Keane, A. (2006). Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal*, 44:879–891.

Knowles, J. (2006). Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

Krettek, J., Schauten, D., Hoffmann, F., and Bertram, T. (2007). Evolutionary hardware-in-the-loop optimization of a controller for cascaded hydraulic valves. In *Proceedings of the IEEE/ASME 2007 International Conference on Advanced Intelligent Mechatronics*, pp. 1–6.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature, PPSN I*, pp. 193–197. Springer Lecture Notes in Computer Science.

Larranaga, P., and Lozano, J. (2001). *Estimation of distribution algorithms: A new tool for evolutionary computation*. Dordrecht, The Netherlands: Kluwer.

Poloni, C., Giurgevich, A., Onesti, L., and Pediroda, V. (2000). Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4):403–420.

Ponweiser, W., Wagner, T., Biermann, D., and Vincze, M. (2008). Multiobjective optimization on a limited budget of evaluations using model-assisted $\mathcal{S}$-metric selection. In *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, pp. 784–794.

Schwefel, H.-P. (1981). *Numerical optimization of computer models*. Chichester, UK: John Wiley.

Ulmer, H., Streichert, F., and Zell, A. (2003). Evolution strategies assisted by Gaussian processes with improved preselection criterion. In *Proceedings of the IEEE Congress on Evolutionary Computation, 2003. CEC '03*, Vol. 1, pp. 692–699.

Ulmer, H., Streichert, F., and Zell, A. (2004). Evolution strategies with controlled model assistance. In *Proceedings of the IEEE Congress on Evolutionary Computation, 2004. CEC2004*, Vol. 2, pp. 1569–1576.

Zhang, Q., Zhou, A., and Jin, Y. (2008). RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63.

Zhou, Z., Ong, Y. S., Nguyen, M. H., and Lim, D. (2005). A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 2832–2839.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, ETH Zurich, Switzerland.

Zitzler, E., and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—A comparative case study. In *Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN V)*, pp. 292–301.