Study of the Approximation of the Fitness Landscape and the Ranking Process of Scalarizing Functions for Many-objective Problems*

Gregorio Toscano
CINVESTAV-IPN
Cd. Victoria, Tamaulipas, Mexico
gtoscano@tamps.cinvestav.mx

Kalyanmoy Deb
Michigan State University
East Lansing, MI, USA
kdeb@egr.msu.edu

COIN Report Number 2016018

Abstract

Although surrogate models have been successfully adopted by evolutionary algorithms to solve time-consuming multiobjective problems, their use has been confined to solving problems with a low number of objectives. On the other hand, scalarizing functions have proved to work well with many-objective problems.

This paper presents a novel study on many-objective optimization concerning the use of surrogate models to approximate both (1) the fitness landscape of traditional multiobjective approaches and (2) the ranking relation imposed by such approaches.

Our methodology involves a thorough comparison of four popular surrogate modeling techniques in order to approximate the fitness landscape and the ranking relations of three different scalarizing functions. Additionally, we explored the interactions of these methods through four well-known scalable test problems with four, six, eight, and ten objectives.

Besides finding that Tchebycheff scalarizing function and Gaussian processes for machine learning are accurate approaches to handle many-objective problems, one of our most important findings involves the capabilities of metamodeling techniques to approximate the ranking procedure from the information gathered from the parameter space. Such a capability can be effectively used for pre-screening purposes on MOEAs.

1 Introduction

The area of evolutionary multiobjective optimization (EMO) has developed a number of methods to solve problems that involve two or more objectives (MOPs). However, previous studies [1–3] have shown that these methods tend to deteriorate in performance when solving problems with more than three objectives (the so-called many-objective

*Accepted for presentation at WCCI-2016 (25-29 July 2016), Vancouver, Canada.

problems [4, 5]). Additionally, although multiobjective evolutionary algorithms (MOEAs) can provide results even for dificult MOPs, they can fail when only a handful of fitness function evaluations are available to produce acceptable results. Comparatively, surrogate models[1] have been recurrently adopted by the EMO community to reduce the required number of fitness function evaluations for time-consuming problems [6–8].

Scalarizing functions are special fitness assignment methods that restate the MOP as a single-objective problem [9]. These functions perform their search on a direction defined by a weight vector. Therefore, a set of weight vectors are required in order to deliver different solutions for a given MOP. Recently, the term decomposition [10] has emerged as a way to adapt scalarizing functions to population-based MOEAs. Decomposition-based MOEAs have been found to be effective when solving many-objective problems [11].

Although the use of scalarizing methods (for solving many-objective problems) and surrogate models (for handling time-consuming MOPs) have been popular trends in the EMO area, more studies about their interaction are still required by the community.

In this regard, MOEAs have incorporated surrogate models to solve problems with three or more objectives in different ways, such as fitness replacement [12, 13], pre-screening solutions [8, 14–17], or as intelligent operators [7, 8, 13]. Most of these approaches have used these metamodels to approximate the fitness landscape of scalarizing functions [7, 15, 16, 18] as well as some performance measures [8, 14, 17], while only a few approaches have opted to surrogate directly the problem at hand [12]. According to these works, it seems that researchers have opted to avoid the direct approximation of each of the functions for a many-objective problem. This seems reasonable, since each approximation adds up its own modeling error, and the metamodeling process will end up having a notable inaccuracy. Furthermore, although decomposition-based approaches solve a set of single objective optimization subproblems, the independent nature of the subproblems prevents them from adding up the errors when the subproblems are approximated by metamodeling techniques. However, the overhead of producing surrogate models for each subproblem can be high.

In spite of the variety of scalarizing functions [9] (as well as other multiobjective fitness assignment methods [19]), researchers that work with metamodels have typically selected the Tchebycheff function [15, 16]. However, since each fitness assignment method produces a different fitness landscape, it is possible that a specific scalarization function can produce a smooth fitness landscape that can be easily approximated by a metamodeling technique.

This work will study the performance of surrogating the fitness landscape of scalarizing functions. Furthermore, since pre-screening solutions have been recurrently used by the community, this study will also address whether the surrogate model is able to maintain the preference relations imposed by the scalarizing function, such that we can rank the predicted solutions and we can obtain a similar order as if they would have been evaluated with the

[1] We will use the terms *approximation models*, *surrogate models*, and *metamodels* interchangeably in this paper.

original function.

Additionally, we want to take the use of surrogate models one step further in the development of MOEAs for many objective problems, since we want to explore whether a surrogate model is able to learn the process of ranking solutions by itself (without approximating the scalarizing function). Albeit this method can be seen as naïve for its simplicity, a successful approach would positively impact the design of new surrogate-based MOEAs for many-objective problems.

To perform our comparative study, we have selected the decision tree regressor (DTR), kernel ridge regression (KRR), Gaussian processes for machine learning (GP), and Bayesian ridge regression (BRR). Similarly, the following scalarizing functions were selected: Tchebycheff (TCH), weighted sum (WS), and penalty boundary intersection (PBI). Additionally, we explored the interactions of these methods through four well-known scalable test problems with four, six, eight, and ten objectives.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of mutiobjective concepts, scalarizing functions, and surrogate modeling techniques adopted in this work. Section 3 presents the state of the art of surrogate-based MOEAs when solving three or more objectives. Section 4 presents our experimental setting and our adopted methodology for fine-tuning the parameters of the studied metamodeling approaches. We divided our experimentation into three main experiments. Section 4.2 introduces our first experiment. This experiment is intended to evaluate whether a scalarizing function produces an easy to approximate fitness landscape and how well the metamodels can fit it. Our second experiment, presented in Section 4.3, measures the efficiency of the compared approaches. Our third experiment, which is shown in Section 4.4, tries to identify those approaches that can be used as pre-screeners. Finally, Section 5 summarizes our main conclusions and outlines future work.

2 Background

2.1 Multiobjective optimization

A multiobjective optimization problem can be defined as the problem of minimizing an objective vector $\boldsymbol{f}(\boldsymbol{x}) = [f_1(x), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})]^T$, where $f_i : \mathcal{F} \rightarrow \mathbb{R}$ is the i-th objective function, $i \in \{1, 2, \ldots, k\}$. The goal is to find a set of Pareto-optimal solutions $\mathcal{P}^* \subset \mathcal{F}$, such that $\mathcal{P}^* = \{\boldsymbol{x}^* \in \mathcal{F} \mid \nexists \boldsymbol{x} \in \mathcal{F} : \boldsymbol{x} \prec x^*\}$. The symbol "$\prec$" denotes the Pareto-dominance relation, which is given by:

$$\boldsymbol{x} \prec \boldsymbol{y} \quad \Leftrightarrow \quad \begin{array}{l} \forall i \in \{1, 2, \ldots, k\} : f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y}) \\ \exists j \in \{1, 2, \ldots, k\} : f_j(\boldsymbol{x}) < f_j(\boldsymbol{y}) \end{array} \quad (1)$$

If $\boldsymbol{x} \prec \boldsymbol{y}$, then $\boldsymbol{x}$ is said to dominate $\boldsymbol{y}$. Otherwise $(\boldsymbol{x} \nprec \boldsymbol{y})$, $\boldsymbol{y}$ is said to be nondominated with respect to x. The image of $\mathcal{P}^*$ in the objective space is called the Pareto-optimal front.

Given a multiobjective optimization problem, we can restate the original formulation in order to produce an scalarizing problem. The comunity of operations research has proposed more than 30 approaches [9].

In this paper, we will focus on the use of Tchebycheff, penalty boundary intersection, and weighted sum methods.

- Tchebycheff (TCH): This method combines multiples objectives into a single objective using the following formulation Minimize $g_{tch}(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{z}^*) = \max_{1 \leq i \leq m} \{w_i |f_i(\boldsymbol{x}) - z_i^*|\}$., where $\boldsymbol{w} = (w_1, \ldots, w_m)$ is a weight vector and $\boldsymbol{z}^* = (z_1^*, \ldots, z_m^*)$ is a reference point in the objective space, where $z_i^* = \min_{\boldsymbol{x} \in X} \{f_i(\boldsymbol{x})\}$, for $i \in \{1, \ldots, m\}$.

- Boundary intersection approach (PBI): This approach [20] uses a weight vector $\boldsymbol{w}$ and a penalty value θ to minimize both, the distance to a reference point d_1 and the direction error to the weighted vector d_2 from the objective values of the solution $\boldsymbol{x}$. The PBI approach can be formulated as follows:

  Minimize $d_1 + \theta d_2$, subject to $\boldsymbol{x} \in \mathcal{Q}$, where $d_1 = \frac{||\boldsymbol{f}(\boldsymbol{x}) - z^*||}{||w||}$ and $d_2 = \left|\left|(\boldsymbol{f}(\boldsymbol{x}) - z^*) - d_1 \frac{w}{||w||}\right|\right|$

- Weighted sum approach (WS): This method expresses the quality of solutions by adding all the objective functions together using weighting coefficients. Thus, the fitness of a solution $\boldsymbol{x}_i$ is given by: $\mathrm{WS}(\boldsymbol{x}_i) = \sum_{m=1}^{M} w_m f_m(\boldsymbol{x}_i)$ where w_m is the weighting coefficient denoting the relative importance of the m-th objective.

2.2 Surrogate models

A surrogate model is a method that aims to simulate the behavior of expensive processes in order to produce simpler and cheaper models. The internal behavior of original function does not need to be known, since surrogate models are usually interested in reproducing only the input/output relation.

- Kernel ridge regression (KRR) is a special case of the more known support vector regression (SVR) whose loss function is the main difference, since it uses a squared error loss while SVR uses ϵ-insensitive loss. However, KRR can be solved using a close form, which translates to faster training times in medium problems. For further details we encourage the reader to review [21].

- Gaussian Models (GM) are a spatial prediction method based on minimizing the mean squared error. GM describes the spatial and temporal correlation among the values of an attribute. In this paper, we have selected the Gaussian processes for Machine Learning (GP) approach.

  In this paper, we have selected the Gaussian processes for Machine Learning (GP) approach.

- Generalized linear models has been consolidated as an adaptive approach that can include regularization of theirs parameters. The Bayesian ridge regression (BRR) [22] is based on Bayesian regression. It estimates the following probabilistic model $p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}\boldsymbol{I_p})$, where the prior for the parameter w is given by a spherical Gaussian.

  The priors over α and λ are chosen to be gamma distributions, the conjugate prior for the precision of the Gaussian.

- Tree-based regression approaches comprises a set of non-parametric supervised learning methods employed for classification and regression. DTs create models that predict the value of a target variable by learning simple decision rules inferred from the data features. Decision tree regressor (DTR) belongs to the DT paradigm and is able to fit functions with addition noisy observation.

AdaBoost is an algorithm that sequentially fits "weak" classifiers and regressors to different weightings of the observations in a dataset. The idea behind the approach is to improve the weight of those observations poorly classified. Empirical evidence has shown that although base classifiers can be fairly simplistic, can capture complex decision boundaries when they are boosted with Adaboost. [23].

3 Surrogate models in multiobjective evolutionary algorithms

Although MOEAs have incorporated surrogate models to solve an important number of difficult MOPs, only a few approaches have solved problems with three or more objectives.

Knowles [18] proposed one of the first surrogate-based MOEAs. The approach is the result of a modification performed on the efficient global optimization (EGO) algorithm [24] in order to enable it to handle multiobjective problems. The augmented Tchebycheff function was used to restate the MOP into a scalarizing one.

Goel et al. [12] incorporated a response surface model to a modified NSGA-II in order to solve a four-objective liquid-rocket injector design problem. Regardless of the number of objectives, this approach used Pareto dominance.

Pilat and Neruda [8] solved problems with a high number of objectives through the use of local aggregate meta-models in a memetic operator (up to 15 objectives). It is worth noting that their approach did not try to surrogate original objective functions, but the distance to the non-dominated points stored in an external file.

Although Emmerich et al. did not solve problems with more than three objectives in [14], their approach is able to handle many-objective problems since it is based on the dominated hypervolume measure. This approach uses a local Gaussian random field metamodel to screen the solutions that increase the estimated hypervolume of the current front.

Bittner and Hahn [17] adopted a similar idea, since their method surrogates the growth of the dominated hypervolume with a Kriging algorithm in a multiobjective particle swarm optimization algorithm. Their approach selects the highest ranked particles to be assessed by the real objective function.

Arias-Montano et al. [16] proposed the use of several surrogate models to optimize airfoil aerodynamic designs. The approach selects the surrogate model with the highest accuracy to apply it at any given time. The algorithm evaluates in the real function only those solutions that minimize the surrogated Tchebycheff function (on a specific weight).

Zhang et al. [15] proposed the MOEA/D-EGO approach, which is a hybrid of MOEA/D and EGO algorithms. The approach uses EGO to surrogate the solutions of the previous iteration. Then, it uses MOEA/D to generate a set of candidate test points. Only a few solutions are selected with respect to their expected improvement for real function evaluation. It is worth noting that this approach was able to solve problems with up to six objective functions.

4 Experimentation

In this paper, we want to evaluate the performance of different metamodeling techniques and scalarizing functions in order to identify those approaches that can be used to screen solutions in MOEAs. To achieve such a goal, the selected metamodeling techniques will be tested on two different problems: the approximation of the fitness landscape of scalarizing methods, and the approximation of the process of ranking solutions. For the sake of brevity, we will refer to our problems as the landscape approximation problem (LAP), and ranking approximation problem (RAP), respectively.

Also, instead of separating the problems in different discussions, we will perform a side-by-side comparison of results. However, since the approaches will solve different problems, we will discuss separately the main goals for each one.

Beside fine-tuning the parameters of the adopted metamodeling approaches, we gathered our research from three main experiments.

The first experiment in LAP is intended to identify both (1) whether the fitness landscape of the scalarizing functions ease the surrogate process, and (2) what metamodeling technique can properly fit such a landscape. Although one could think that in RAP the scalarizing function does not play an important role, we think that since the ranking procedure is performed after the scalarizing functions have assigned the fitness values to each solution, they indeed contribute to the ranking process. With this in mind, we should expect some differences when testing different scalarizing functions. We want to evaluate the impact of the use of the scalarizing functions to the ranking process, as well as identify the metamodeling technique that better predicts such a ranking.

Our second experiment will reveal the computational efficiency of our compared surrogate models.

Finally, our third experiment will evaluate the ability of each pair (metamodeling technique, scalarizing function) to be used as pre-screening approaches.

To assess our methods we have selected four well-known problems: DTLZ1, DTLZ2, DTLZ3, and DTLZ4 [25]. These problems can be scaled to any number of objectives and decision variables. The total number of variables in these problems is $n = M + k - 1$, where M denotes the number of objectives and k is a difficulty parameter. We used $k = 5$ for DTLZ1 and $k = 10$ for the remaining problems. Experiments were performed for instances with $M \in \{4, 6, 8, 10\}$ objectives for the four considered test problems.

We fed our experiments with datasets generated as follows:

1. Since the benefits of building local surrogates on smaller regions have been clearly addressed by different works [8, 26–29], we decided to simulate the use of local surrogate models by restricting the generation of solutions to one tenth of the whole range of each variable. The position of the subrange was randomly selected.

2. Our training dataset for every execution comprises 30 different sets with 100 points

randomly generated on the range previously explained. The testing dataset is of the same size of our training dataset.

4.1 Parameter tuning

Since the selection of the parameters of a metamodeling technique plays a key role in their performance, we decided to perform a parameter tuning before using such methods in our experiments. Rather than measuring the accuracy of the approximations, we are seeking to identify parameters that induce a low variance in the response of the methods (maintaining the accuracy), since variance can better estimate the behavior of these models when handling unseen data [30] (assuming that bias affects to all approaches similarly). Therefore, we are interested in reducing the over-fitting of our adopted metamodeling techniques.

Researchers have used cross validation (CV) as a way to measure the predictive performance of statistical models. However, even when sufficient data is available for training and testing the model, not all CV variants reduce the chances to produce overfitted models. K-fold CV [30] has been successfully used to validate surrogate models.

In this work, we have adopted K-fold CV to reduce the variance of the studied approaches. This methodology is described next:

First, we need to perform a manual binning of the continuous parameters and perform a full factorial design of the most widely-used parameters. Then, we will divide the dataset into $k = 10$ subsets, each one of these subsets will serve to validate the model, while the other $k - 1$ remaining subsets will be gathered together to form a training dataset. This procedure is required to be performed $k = 10$ times (every subset must be selected to validate the model). The error of the approach is computed as the average error across all k trials. Although the variance of the estimate can be reduced when k is increased, using large values of k translates into a very time consuming method. Notwithstanding, $k = 10$ has been found to produce acceptable results for our purposes [30].

Below, we present the parameters selected for the full factorial design.

- GP: $\theta_L \in$ {0.001, 1e-4, 1e-6}, $\theta_0 \in$ {0.,0.01, 0.001}.
- KRR: $\alpha \in${1, 0.1, 0.01, 0.001}, $\gamma \in$ {0.01, 0.1, 1, 10, 100}.
- BRR: $\alpha_1 \in$ {0.01, 0.0001, 1e-6, 1e-8}, $\alpha_2 \in$ {0.01, 0.0001, 1e-6, 1e-8}, $\lambda_1 \in$ {0.01, 0.0001, 1e-6, 1e-8}, $\lambda_2 \in${0.01, 0.0001, 1e-6, 1e-8}.

Below, we give our undertaken methodology to perform the fine-tuning procedure:

1. 30 datasets with 100 points each for every test problem was created using Latin hypercubes[2] [31] (a size of 100 was selected since most MOEAs typically handle this population size).
2. The K-fold CV methodology was applied to each approach on each of the 30 datasets built in the previous step (for all the problems).
3. In spite of homogeneous selection of parameters that produced the K-fold CV

[2] Statistical method of stratified sampling that can be applied to multiple variables

methodology, we could observe some variations when solving different test functions. Therefore, we left the selection process to a central tendency statistics: Mode (we selected the values that lied on the mode).

Below we present the selected parameters after performing the fine-tuning procedure.

- BRR: $\alpha_1 = 1e - 8$, $\alpha_2 = 0.01$, $\lambda_1 = 0.01$, $\lambda_2 = 0.01$.
- DTR: use Adaboost for model improvement. Adaboost used 300 estimators to improve DTR's predition.
- GP: corr = CUBIC, $\theta_0 = 0.1$, $\theta_L = 0.001$, $\theta_U = 1e - 1$.
- KR: kernel = RBF, $\gamma = 10.0$, $\alpha = 0.001$.

4.2 Experiment 1

As previously mentioned, in this experiment we want to identify both whether the fitness landscape of the scalarizing functions ease the surrogate process and what metamodeling technique can fit properly such a landscape (in the LAP). This experiment will also help us to evaluate the impact of the use of a scalarizing function to the ranking process, and to identify the metamodeling technique that predicts better such a ranking.

Below, we summarize the elements of our designing set:

- Metamodels = {DTR, KRR, GP, BRR}).
- Scalarizing functions = {TCH, WS, PBI}.
- MOPs = {DTLZ1, DTLZ2, DTLZ3, DTLZ4}.
- Objectives = {4,6,8,10}.

For each metamodel, scalarizing function, MOP, problem size (measured for the number of objectives), we performed 30 different executions (a total of 5,760 executions).

For each execution in LAP, we fed the scalarizing function at hand with the training and testing datasets (with 100 points each). The solutions of the training dataset and their associated fitness computed by the scalarizing function fed the training process of the metamodeling at hand. It is important to mention that in this study we only evaluate the use of one single weight vector, then, the trained surrogate model is used to predict the testing dataset. We assessed the performance of the pair (metamodeling, scalarizing function) using the the standard error, R^2, and MSE metrics. However, due to space limitations, we only show the results of the standard error metric.

RAP executions were similar to the LAP ones, with the exception that after computing the training dataset on the scalarizing function, a ranking process was called in order to give an ordered index to each solution. Such set of indexes fed the metamodeling approaches. We assessed the performance of the pair (metamodeling, scalarizing function) using the the standard error metric.

The standard error measures the accuracy of predictions using $\sigma_{est} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$ where n is the size of the validation dataset, $\hat{y}_i$ is the predicted value for the input i, and y_i is the real value; $\bar{y}$ is the mean of the real values.

Figures 1(a) and 1(c) present the results produced in LAP, whereas Figures 1(b)and 1(d) present the results produced in RAP. The box-plots shown in Figure 1 suggest different conclusions for LAP and RAP. The standard error gathered in LAP consistently ranks TCH as the scalarizing function that induced better outcomes regarding the metamodeling techniques at hand. WS also induced a good behavior on the modeling approaches. However, it remains far from the results induced by TCH. Finally, PBI stays far behind TCH and WS. Figure 1(c) shows the effects when the dimensionality of the problems was raised. In this figure, it is clear that the variance of the approaches worsen as the number of objectives raises. However, from these graphics it is clear that the GP approach steps ahead of the others.

Figure 1(d) shows quite a different behavior when the approaches are solving RAP. Although it is possible to appreciate a slight worsening trend, the variance of the approximations remains almost inalterable regardless of the raise in the number of objectives. This seems quite natural, since the n-dimensional problem is restated to a 1-dimensional problem. Therefore, the raise in the objectives does not affect the surrogate models in this problem.

Contrary to the results obtained in LAP, TCH systematically remains behind the other two approaches in RAP (see Figure 1(b)). WS was the approach with the best median in this problem.

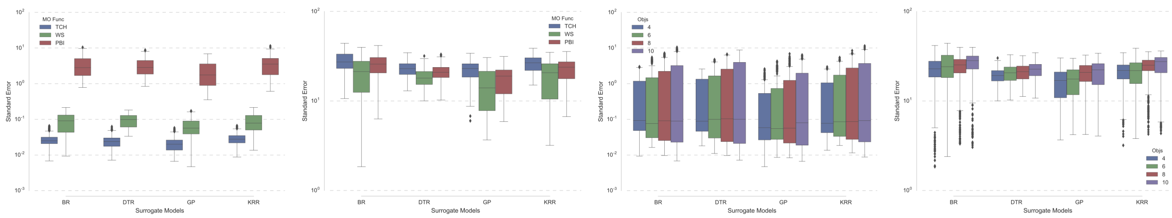
The results presented above suggest that the TCH induces a better behavior on the compared metamodeling techniques since it induced a raise in their accuracy, regardless the problem at hand. It is worth noting that TCH promotes surrogate modeling techniques that improve in their robustness as well. After analyzing the box-plots graphics shown in Figure 1,we can argue that GP behaved slightly better than the other approaches, since it produced medians closer to 0 and its variance was marginally better than the rest.

On the other hand, when the studied approaches tried to solve RAP, all the approaches presented a very constant scalability. However, we can also observe that RAP affects the accuracy of the metamodeling approaches.

4.3 Experiment 2

Efficiency is an important issue in surrogate models since we want to approximate an expensive function with a cheaper model. Therefore, we will prefer those metamodeling techniques with a lower overhead (but maintaining the accuracy). Our second experiment measures the efficiency of the compared approaches. The experiments were carried out in a MacBook Air computer running OSX 10.11.3 on a 1.7GHz Intel Core i7 processor with 8 MGB 1600 MHZ DDR3 of RAM. The implementations were undertaken in Python 2.7.11 using numpy and sklearn libraries.

We followed the same experimental setup explained in Section 4.2; however, in this experiment, we measured the time employed to construct a surrogate model (with 100 points) and the required time that it takes to predict 100 other responses. The results of this experiment are shown by means of violin plots in Figure 2. Violin plots are able to compare the distribution of quantitative data across levels of categorical variables. These plots widen their body in more populated regions while presenting a thin body in less clustered regions. Figures 2(a) and 2(b) show the time consumed by the studied approaches when



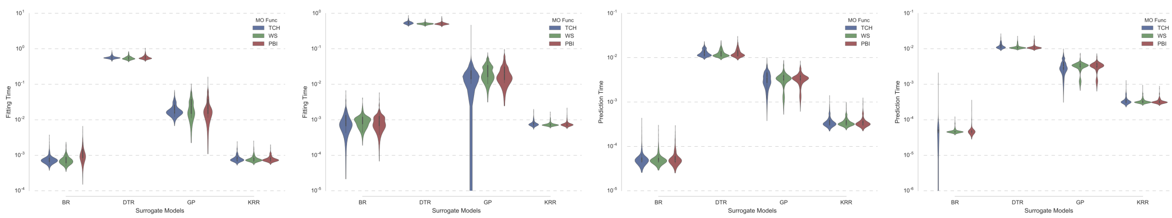
(a) (b) (c) (d)

Figure 1: Experiment 1: Results of the execution of standard error metric. box-plots shown in Figures (a) and (c) show the information gathered for LAP, whereas Figures (b) and (d) show the information produced in RAP.

solving LAP and RAP, respectively. Both graphics convey the same information indicating that each surrogate model required a very homogeneous time regardless the scalarizing function they were approximating. From these graphics it is clear that DTR was the less efficient approach, followed by GP. The high computational time required by DTR is thoroughly understandable, since this approach runs a time-consuming performance booster. On the other hand, KRR and BR behaved quite similar. Both approaches consistently required the lowest amount of time to construct the surrogate model. The results involving the required time to predict solutions are shown in Figures 2(c) and 2(d) for LAP and RAP, respectively. These results convey with the same information as the produced when constructing the surrogate models, although the predicted task involved less time. Even when one would expect less overhead in RAP, the results disclose the opposite.

4.4 Experiment 3

We found that several works that use surrogate models in evolutionary algorithms embraced the idea of evaluating only those higher ranked solutions. Screening solutions seem to be a natural approach to incorporate surrogate models into MOEAs since it resembles the natural selection principle, in which only those individuals that compete most effectively for resources will survive for the next generation. In this regard, we are interested in evaluating whether the surrogate model can capture the preference relations that the scalarizing function imposes over its evaluated solutions. Then, we would like to detect any discrepancy in the preference relations proposed by the surrogate model. To achieve our goal, we undergo the same experimental setup as the one explained in Section 4.2. We will give an overall explanation of the process. We trained our surrogate models with 100 solutions uniformly distributed in a clustered region of the search space. Then, we assessed 100 randomly generated solutions in the scalarizing function at hand and we asked the surrogate



(a) (b) (c) (d)

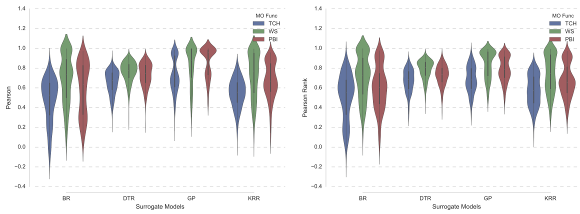
Figure 2: Experiment 2: Results of measuring the training and prediction times required by the metamodels. Figures (a) and (c) contain information regarding to the training time of the metamodels for LAP, whereas Figures (b) and (d) contain information regarding to the training time of the metamodels for RAP.

model to predict the outcome. We then computed the Pearson correlation coefficient of the expected solution and the predicted one (we also computed the Kendall and Spearman coefficients, but we did not find any discrepancy between the measures, so we opted for presenting only one metric). Such a process is performed 30 independent times for each configuration studied (surrogate models, scalarizing functions, MOPs, and number of objectives). A Pearson correlation coefficient's value of one would indicate that the surrogate model could capture the preference relations imposed by the scalarizing function. Approaches that consistently show a high value in this metric can be used to screen solutions in the evolutionary process. In this study we are not interested in achieving accurate solutions, but rather a good precision in the preference relations among solutions.

Replicating this experiment in our RAP required an additional step that is explained as follows. Before providing the validation set to the previously trained surrogate model, we sorted the solutions according to their fitness in the scalarizing function. Therefore, we expected an ordered outcome from the surrogate model. It is worth noting that in order to avoid imposing any bias to the surrogate models, we fed the training process with randomly generated solutions (with no order imposed).

Figures 3(a) and 4(a) shows the obtained results for the landscape approximation. We found unexpected results in LAP, since the scalarizing function that induced the most accurate results over the surrogate models (shown in Section 4.2) was the method with the worst results in maintaining the rations imposed by itself: TCH. GP out of the remaining surrogate modeling techniques could produce acceptable results for problems with four objectives. On the other hand, multiobjective methods that induced a less accurate behavior on the surrogate modeling techniques such as WS and PBI induced good performance on the surrogate models, being GP the metamodeling approach that could make the best surrogates for both scalarizing.

The results achieved for the RAP are quite similar to the ones obtained in LAP (please refer



(a) (b)

Figure 3: Experiment 3: Figure (a) contains a summarized information regarding to the Pearson Coefficient Correlation measured for LAP, whereas Figures (b) contains a summarized information regarding to the Pearson Coefficient Correlation measured for RAP.

to Figures 3(b) and 4(b)).

These results are very encouraging since surrogate models were able to replicate the ranking process. The results are also interesting in the sense that even when a scalarizing function can induce a higher accuracy when it is approximated by a surrogate model, it cannot guarantee to maintain the preference relations among solutions. Therefore, we should opt for methods that comply with the preference relations among solutions rather than select the ones that achieve accurate predictions. Of course, methods that meet both characteristics would be better.

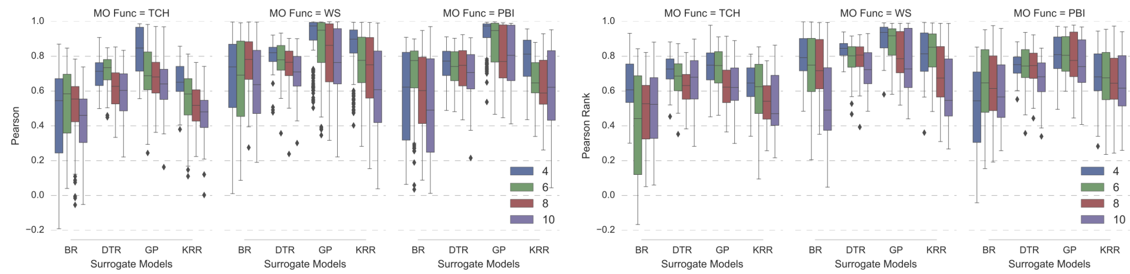
5 Conclusions and future work

This paper presents a novel study on many-objective optimization about the use of surrogate models to approximate both (1) the fitness landscape of traditional multiobjective approaches and (2) the ranking relation imposed by such multiobjective approaches.

Let us first discuss the achievements obtained in our first problem (fitness landscape approximation problem). Among the results produced by this study, we observe that TCH eases the approximation task of the surrogate modeling techniques since all of them produced their most accurate predictions with this scalarizing function. On the other side, when surrogate models were handling the approximation of PBI, their obtained results were not accurate at all (with respect to the results obtained with TCH). Moreover, among the different surrogate modeling approaches, GP stands out from the others, since it consistently produced better results. In general the approaches did not suffer the curse of the dimensionality. Nonetheless more research should be done in this regard. These results suggest that GP and TCH would be an easy pick whether accuracy is a main concern.

When the approaches tried to solve RAP, the results drew a very different scenario. TCH was not even close to its performance in LAP. In this regard, WS was the technique that induced the best behavior of the surrogate models, followed by a close PBI. In this problem, the GP metamodeling technique produced the best results.

When we analyzed the efficiency of the surrogate modeling approaches in terms of the computational time associated with the construction of the model and prediction of solutions, we found that BR and KRR produced almost constant times regardless the multiobjective method that they were trying to approximate. GP produced good results just behind BR and KRR, while DTR was the most time-consuming approach. DTR results were expected since we approached it with a performance boosting approach.



(a) (b)

Figure 4: Experiment 3: Figure (a) contains a detailed information regarding to the Pearson Coefficient Correlation measured for LAP, whereas Figures (b) contains a detailed information regarding to the Pearson Coefficient Correlation measured for RAP.

Finally, the correlation experiment produced unforeseen results. It found that even when the fitness landscape of TCH induced the most accurate results on the metamodeling approaches, they could not assess the ranking relation imposed by the scalarizing function. On the other hand, WS and PBI who behaved mediocrely in the accuracy experiment, outperformed TCH.

Besides finding that TCH and GP are accurate approaches to handle many objective problems, one of our more important achievements refers to the capability of metamodeling techniques to approximate the ranking procedure from the information gathered from the parameter space. This capability can be effectively used for pre-screening purposes in MOEAs. The fact that the raise of the dimensionality in the objective space does not affect this approximation is also interesting, and unlike surrogating the fitness landscape, there is not need to performe one approximation for each vector. This translates into a computational cheaper way to pre-screen solutions even in no very time-consuming many-objective problems.

As part of our future work, we want to extend our experiments to investigate how the inclusion of more weight vectors affect the performance of the metamodeling techniques. In this regard, the efficiency will be compromised in LAP, since it will be necessary to perform one surrogate model for each subproblem produced by the different weights.

Also, even when the error of the subproblems are independent, it can be a reduction in the ranking process, because this contemplates the outcome of every solution with respect to the weight vector. In this regard, our second approach (RAP) can maintain their results, since it will not be affected by the set of weight vectors. Finally, we would like to perform a proof of concept into a MOEA.

Acknowledgements

The first author would like to thank Prof. Kalyanmoy Deb and Michigan State University

for accepting him as a visiting scholar from 08/01/2015 to 07/31/2016.

References

[1] R. C. Purshouse and P. J. Fleming, "Evolutionary Multi-Objective Optimisation: An Exploratory Analysis," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 3. Canberra, Australia: IEEE Press, December 2003, pp. 2066–2073.

[2] V. Khare, X. Yao, and K. Deb, "Performance Scaling of Multi-objective Evolutionary Algorithms," in *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds. Faro, Portugal: Springer. Lecture Notes in Computer Science. Volume 2632, April 2003, pp. 376–390.

[3] E. J. Hughes, "Evolutionary Many-Objective Optimisation: Many Once or One Many?" in *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, vol. 1. Edinburgh, Scotland: IEEE Service Center, September 2005, pp. 222–227.

[4] A. Jaszkiewicz, "On the Performance of Multiple-Objective Genetic Local Search on the 0/1 Knapsack Problem—A Comparative Experiment," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, August 2002.

[5] E. J. Hughes, "Multiple Single Objective Pareto Sampling," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 4. Canberra, Australia: IEEE Press, December 2003, pp. 2678–2684.

[6] Y. Jin, B. Sendhoff, and E. Körner, "Evolutionary Multi-objective Optimization for Simultaneous Generation of Signal-Type and Symbol-Type Representations," in *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Guanajuato, México: Springer. Lecture Notes in Computer Science Vol. 3410, March 2005, pp. 752–766.

[7] A. Díaz-Manríquez, G. Toscano-Pulido, and R. Landa Becerra, "A Surrogate-based Intelligent Variation Operator for Multiobjective Optimization," in *International Conference on Artificial Evolution*, France, 2012.

[8] M. Pilát and R. Neruda, "Aggregate meta-models for evolutionary multiobjective and many-objective optimization," *Neurocomputing*, vol. 116, pp. 392–402, September 20 2013.

[9] K. Miettinen and M. M. Mäkelä, "On scalarizing functions in multiobjective optimization," *OR Spectrum*, vol. 24, no. 2, pp. 193–213. [Online]. Available: http://dx.doi.org/10.1007/s00291-001-0092-9

[10] Z. Zhang, "Immune optimization algorithm for constrained nonlinear multiobjective optimization problems," *Applied Soft Computing*, vol. 7, no. 3, pp. 840–857, June 2007.

[11] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, December 2007.

[12] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, "Response surface approximation of Pareto optimal front in multi-objective optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4-6, pp. 879–893, 2007.

[13] D. Chafekar, L. Shi, K. Rasheed, and J. Xuan, "Multiobjective GA optimization using reduced models," *IEEE Transactions on Systems Man and Cybernetics Part C–Applications and Reviews*, vol. 35, no. 2, pp. 261–265, May 2005.

[14] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, August 2006.

[15] Y. Zhang, Y. Jun, G. Wei, and L. Wu, "Find multi-objective paths in stochastic networks via chaotic immune PSO," *Expert Systems with Applications*, vol. 37, no. 3, pp. 1911–1919, March 15 2010.

[16] A. Arias-Montano, C. A. Coello Coello, and E. Mezura-Montes, "Multi-objective Airfoil Shape Optimization using a Multiple-surrogate Approach," in *IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8.

[17] F. Bittner and I. Hahn, "Kriging-Assisted Multi-Objective Particle Swarm Optimization of Permanent Magnet Synchronous Machine for Hybrid and Electric Cars," in *2013 IEEE International Electric Machines & Drives Conference (IEMDC 2013)*. Chicago, Illinois, USA: IEEE Press, May 12-15 2013, pp. 15–22, iSBN 978-1-4673-4974-1.

[18] J. Knowles, "ParEGO: A Hybrid Algorithm with On-line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, January 2006.

[19] M. Garza-Fabre, G. T. Pulido, and C. A. Coello Coello, "Alternative Fitness Assignment Methods for Many-Objective Optimization Problems," in *Artifical Evolution, 9th International Conference, Evolution Artificielle, EA 2009*. Strasbourg, France: Springer. Lecture Notes in Computer Science, Vol. 5975, 2010, pp. 146–157, iSBN 978-3-642-14155-3.

[20] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[21] G. Box and N. Draper, *Response Surfaces, Mixtures, and Ridge Analyses*. John Wiley & Sons, 2007, vol. 649.

[22] D. J. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1991.

[23] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157.

[24] D. Jones, M. Schonlau, and W. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *J. of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998.

[25] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multiobjective Optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. USA: Springer, 2005, pp. 105–145.

[26] A. Isaacs, T. Ray, and W. Smith, "A Hybrid Evolutionary Algorithm With Simplex Local Search," in *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*. Singapore: IEEE Press, September 2007, pp. 1701–1708.

[27] H. Liu and J. Li, "A particle swarm optimization-based multiuser detection for receive-diversity-aided STBC systems," *IEEE Signal Processing Letters*, vol. 15, pp. 29–32, 2008.

[28] C. Georgopoulou and K. Giannakoglou, "Multiobjective Metamodel–Assisted Memetic Algorithms," in *Multi-Objective Memetic Algorithms*, ser. Studies in Computational Intelligence. Springer Berlin / Heidelberg, 2009, vol. 171, pp. 153–181.

[29] A. Díaz-Manríquez, G. Toscano-Pulido, and W. Gomez-Flores, "On the Selection of Surrogate Models in Evolutionary Optimization Algorithms," in *IEEE Congress on Evolutionary Computation*, june 2011, pp. 2155–2162.

[30] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.

[31] M. McKay, R. Beckman, and W. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.