On the Selection of Surrogate Models in Evolutionary Optimization Algorithms

Alan Díaz-Manríquez, Gregorio Toscano-Pulido and Wilfrido Gómez-Flores
Information Technology Laboratory, CINVESTAV-Tamaulipas
Parque Científico y Tecnológico TECNOTAM
Km. 5.5 carretera Cd. Victoria-Soto La Marina
Cd. Victoria, Tamaulipas 87130, MÉXICO
({adiazm,gtoscano,wgomez}@tamps.cinvestav.mx)

***Abstract*—Since many real-world problems are related to the satisfaction of at least one goal, several optimization techniques have been proposed in the past. However, traditional optimization techniques are computationally expensive and are normally highly susceptible to some characteristics such as high dimensionality, non-differentiability, non-linearity, highly expensive function calculation, among others. Evolutionary algorithms are bio-inspired meta-heuristics that have shown flexibility, adaptability and good performance when solving these sort of problems. In order to achieve acceptable results, some problems usually require several evaluations of the optimization function. However, when each of these evaluations represents a high computational cost, these problems remain intractable even by these meta-heuristics.**

To reduce the computational cost in expensive optimization problems, some researchers have replaced the real optimization function with a computationally inexpensive surrogate model. Despite there are comparison studies among these techniques, these studies focused on revised the accuracy of the meta-model for the problem at hand, but neither its suitability to be used with evolutionary algorithms, nor its scalability in the variable design space.

In this work, we compare four meta-modeling techniques, polynomial approximation, kriging, radial basis functions and support vector regression, in different aspects such as accuracy, robustness, efficiency, and scalability with the aim to identify advantages and disadvantages of each meta-modeling technique in order to select the most suitable one to be combined with evolutionary optimization algorithms.

I. Introduction

In recent years, Evolutionary Algorithms (EAs) have been successfully applied in an important variety of difficult optimization problems. Its main advantage lies in their ability to locate solutions close to the global optimum. However, for many real-world problems, the number of calls to the objective function to locate a near-optimal solution in the EAs may be high. In many science and engineering problems researchers have been used computer simulation approaches in order to replace expensive physical experiments with the aim to improve the quality and performance of engineered products and devices, but using a fraction of the needed effort. For example, Computational Fluid Dynamics solvers, Computational Electro Magnetics and Computational Structural Mechanics have been shown to be very accurate. However, such simulations are often computationally expensive, such that, some simulations can take several days or even weeks in order to be computed. Hence, in many real-world optimization problems, the computational cost is dominated the quantity of objective function evaluations needed to obtain a good solution. Then, it is crucial to use any information gained during the optimization process in order to increase the efficiency of optimization algorithms and consequently, to reduce the number of objective function evaluations.

To build a model of the fitness function to assist in the selection of candidate solutions for evaluation has been a natural approach to utilize the discovered information by the EA. A variety of techniques for constructing the models[1] for computationally expensive optimization problems have been considered in the past [1]–[5]. However, most existing works do not justify why the meta-model was selected.

There are a variety of works where a comparison among different techniques is performed [6]–[9]. However, most of these works make comparisons but only among a small group of techniques and test problems, or they take into account only one criterion for the selection of the best model. Despite, there are only a few works where the meta-modeling technique is chosen according to multiple criteria [10], they usually omit to measure two important factors: dimensionality and the suitability of the meta-model to be combined with evolutionary optimization algorithms.

Hence, in this work, we focus on the evaluation of different aspects of surrogate models: accuracy, robustness, efficiency, simplicity and scalability in order to identify the meta-model to be used with an evolutionary optimization algorithm. The remainder of this work is organized as follows: In Section 2, we give a brief overview of the surrogate modeling techniques commonly used in the literature. Section 3 presents the proposed methodology for the comparison of the meta-modeling techniques. Experimental results obtained on synthetic test problems are presented in Section 4. Finally, Section 5 summarizes our main conclusions as well as some possible directions for future work.

II. Background

A surrogate model is a mathematical model that mimics the behavior of a computationally expensive simulation code

[1]In this work the terms surrogate, meta-model or approximation models will be used interchangeably.

978-1-4244-7833-0/11/$26.00 ©2011 IEEE

over the complete parameter space as accurately as possible, using the least amount of data points. There are a variety of techniques for creating meta-models, such as, rational functions, radial basis functions, artificial neural networks, kriging models, support vector machines, splines, polynomial approximation models. Below, we introduce the most common approaches:

A. Polynomial approximation models

The response surface methodology (RSM) employs the statistical techniques of regression and analysis of variance in order to obtain the responses' minimum variance.

For most response surfaces, the polynomials are used as an approximation function (PRS) for its simplicity, although other types of function can be used.

In general, a polynomial in the coded inputs $x_1, x_2, ..., x_k$ is a function which is a linear combination of powers and products of the $x's$. A term in the polynomial is said to be of degree j if it contains the product of j of the $x's$.

The polynomial model is written in matrix notation as $\hat{y^p} = \beta^T \bar{\mathbf{x}}^p$, where β is the vector of coefficients to be estimated, and $\bar{\mathbf{x}}^p$ is the vector corresponding to the form of the $x_1^{(p)}$ and $x_2^{(p)}$ terms in the polynomial model.

To estimate the unknown coefficients of the polynomial model, both the least squares method (LSM) and the gradient method can be used. In both cases, the number of samples of the real objective function, must be at least as the number of coefficients.

The PRS can be constructed by Full Regression or by Stepwise Regression. The basic procedure for Stepwise Regression involve (1) identifying an initial model, (2) iteratively "stepping", that is, repeatedly altering the model at the previous step by adding or removing a predictor variable in accordance with the "stepping criteria", and (3) terminating the search when stepping is no longer possible given the stepping criteria, or when a specified maximum number of steps has been reached.

In this work, only second degree PRS models are considered.

B. Kriging

Kriging [11] is a spatial prediction method based on minimizing the mean squared error, which belongs to the group of Geo-statistical methods and describing the spatial and temporal correlation among the values of an attribute.

The Design and Analysis of Computer Experiments (DACE) is a parametric regression model developed by *Sack et al.* [12]. DACE is an extension of the Kriging approach in order to be able to manage three or more dimensions.

The DACE model can be expressed as a combination of a know function $a(\mathbf{x})$ (e.g., polynomial function, trigonometric series) and a Gaussian random process $b(\mathbf{x})$, $\hat{y}(\mathbf{x}) = a(\mathbf{x}) + b(\mathbf{x})$. The Gaussian random process $b(\mathbf{x})$ is assumed to have mean zero and covariance:

$$\mathrm{E}(b(\mathbf{x}^{(i)}), b(\mathbf{x}^{(j)})) = \mathrm{Cov}(b(\mathbf{x}^{(i)}), b(\mathbf{x}^{(j)})) = \sigma^2 \mathcal{R}(\theta, \mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \quad (1)$$

where σ^2 is the process variance of the response and $\mathcal{R}(\theta, \mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is the correlation model with parameters θ. Exist different types of correlation models (Exponential, Gaussian, Linear, Spherical, Cubic, Spline).

C. Radial Basis Function Network

The Radial Basis Function method (RBF) was proposed by Hardy [13] in 1971. A RBF is a real-valued function whose value depends only on the distance from the origin, so that $\phi(\mathbf{x}) = \phi(||\mathbf{x}||)$; or alternatively on the distance from some other point c, called a center, so that $\phi(\mathbf{x}, \mathbf{c}) = \phi(||\mathbf{x} - \mathbf{c}||)$. Any function ϕ that satisfies the property $\phi(\mathbf{x}) = \phi(||\mathbf{x}||)$ is a radial function. The norm is usually Euclidean distance, although other distance functions are also possible:

Typical choices for the RBF include linear, cubic, multiquadractics or thin-plate splines, or Gaussian functions.

A RBFN typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. The output, $\varphi : \mathcal{R}^n \to \mathcal{R}$, of the network is thus $\varphi(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(||\mathbf{x} - \mathbf{c}_i||)$.

In a RBF network there are three types of parameters that need to be chosen to adapt the network for a particular task: The weights w_i, the center vector $\mathbf{c}_i$, and the RBF width parameters β_i. In the sequential training of the weights are updated at each time step as data streams.

D. Support Vector Regression

The Support Vector Machines (SVM) draws it inspiration from statistical learning theory [14]. SVM are a set of related supervised learning methods which analyze data and recognize patterns. A support vector machine constructs a hyperplane or a set of hyperplanes in a high dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general, the generalization error of the classifier is opposite to this margin.

SVM schemes use a mapping into a larger space so that cross products may be computed easily in terms of the variables in the original space making the computational load reasonable. The cross products in the larger space are defined in terms of a kernel function $K(x, y)$ which can be selected to suit the problem. In the Table I, show different types of kernel functions.

TABLE I
KERNEL FUNCTIONS FOR SVM

Type of Kernel	Kernel Function
Polynomial	$K(x, x') = \langle x, x' \rangle^d$
Gaussian Radial Basis Function	$K(x, x') = \exp\left(-\frac{\|\|x-x'\|\|^2}{2\sigma^2}\right)$
Exponential Radial Basis Function	$K(x, x') = \exp\left(-\frac{\|\|x-x'\|\|}{2\sigma^2}\right)$
Multilayer Perceptron	$K(x, x') = \tanh(\rho\langle x, x' \rangle + \varrho)$

SVM can also be applied to regression problems[2] by the introduction of an alternative loss function. The loss function must be modified to include a distance measure.

Consider the problem of approximating the set of data with a linear function,

$$f(x) = \langle w, x \rangle + b \tag{2}$$

the optimal regression function is given by the minimum of the function,

$$\phi(w, \xi) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i^- + \xi_i^+) \tag{3}$$

where C is a pre-specified value, and ξ^+,ξ^- are slack variables representing upper and lower constraints on the outputs of the system.

III. Description of the Experiments

Since the main challenge when using approximation models is to be as accurate as possible over the complete domain of interest while minimizing the simulation cost (efficiency), the most of the approaches take into account only the accuracy of the technique [6]–[9]. However, other approaches suggest the use of multiple criteria for assessing the quality of a meta-model [10], i.e., the robustness, the efficiency or the simplicity. Therefore, in order to allow a quantitative assessment of the performance of the meta-model technique, we decided to consider five issues: accuracy, robustness, efficiency, scalability and suitability to be combined with an evolutionary approach.

To evaluate the performance of the different approaches, we selected six scalable unconstrained global optimization test functions described in [15]–[18] from the specialized literature. The test functions chosen were selected based on the shape of the search space and on the number of local minima, i.e., they contain characteristics that are representative of what can be considered "difficult" in global optimization problems. A summary of such features are given in Table II. Below, we describe in detail the general approach used in this work.

1) Perform a full factorial design of the discretized parameters in order to avoid affecting the techniques by a poor tuning of the parameters. The discretized parameters used for each technique are shown below:

Polynomial Regression	Degree of the polynomial=2 Technique for construct the PRS={Full, Stepwise}
Kriging	Correlation function = {Gaussian, Cubic Exponential, Linear, Spherical, Spline}
Radial basis function	Neurons in the hidden layer={3-100}
Support Vector Regression	C ={$2^{-5} : 2^{15}$},γ ={$0.1 : 2$} ϵ ={$2^{-10} : 2^{5}$}

For each meta-modeling technique was chosen the set of parameters which had the best average accuracy in all the problems (a setting for each meta-modeling technique), this is named as **Best overall settings**. In the same way was chosen the set of parameters that had the best accuracy for each problem (54 settings for each meta-modeling technique), this is named as **Best local settings**. The best parameters found using the accuracy for the 54 problems (Best overall settings) are shown below:

Polynomial Regression	Degree of the polynomial=2 Technique for construct the PRS={Stepwise}
Kriging	Correlation function = {Exponential}
Radial basis function	Number of neurons in the hidden layer={6}
Support Vector Regression	C ={$2^{10.5}$}, γ ={0.2}, ϵ ={$2^{-2.5}$}

2) Create a training data set with a latin hypercube[3] [19] of size 100, (we set 100 since an evolutionary algorithm typically handles this number as the population size).
3) Train each technique used (PRS, KRG, RBF, SVR) with the training data set.
4) Create the validation data set with a latin hypercube of size 200.
5) Predict the validation data set with the meta-model.
6) Compute the performance measures for the different adopted criteria.

Accuracy: The accuracy is the capability of the technique to have a prediction close to the real value of the system. In this paper, we use the G-metric in order to measure the accuracy of the meta-model:

$$G = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} = 1 - \frac{\mathbf{MSE}}{\text{Variance}} \tag{4}$$

where N is the size of the validation data set, $\hat{y}_i$ is the predicted value for the input i, and y_i is the real value; $\bar{y}$ is the mean of the real values. The Mean Square Error (MSE) measures the difference between the estimator and the real value. The variance describes how far the values lie from the mean, that is, the variance captures how irregular the problem is. In this metric, we will prefer larger values of G, since this represents a more accurate meta-model.

Robustness: It refers to the capability of the technique to achieve a good accuracy for different test problems. To measure these criteria, we selected six test problems with different features. The problems are shown in Table II. Then, the robustness of a meta-model will be given by its average behavior for all adopted test cases.

Scalability: The scalability is the capability of the technique to achieve a good accuracy for the increasing number of variables in the design space. All the adopted problems can be scaled in the number of decision variables. In order to measure these criteria, we decided to adopt the following problem sizes: $v = [2, 4, 6, 8, 10, 15, 20, 25, 50]$. Then, we will say that a meta-model has a better scalability if it has a good accuracy for all problem sizes.

[2]The SVM for regression problems are know as Support Vector Regression (SVR)

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

TABLE II
FEATURES OF THE TEST PROBLEMS ADOPTED.

Problem name	Space search shape	# of local minima	# variables	Global minimum
Step	Unimodal	no local minima except the global one	n	$x^* = (0, \ldots, 0)$, $f(x^*) = 0$
Sphere	Unimodal	no local minima except the global one	n	$x^* = (0, \ldots, 0)$, $f(x^*) = 0$
Rosenbrock	Unimodal for $n \leq 3$ otherwise multimodal	several local minima for $n > 3$	n	$x^* = (1, \ldots, 1)$, $f(x^*) = 0$
Ackley	Multimodal	several local minima	n	$x^* = (0, \ldots, 0)$, $f(x^*) = 0$
Rastrigin	Multimodal	large number of local minima	n	$x^* = (0, \ldots, 0)$, $f(x^*) = 0$
Schwefel	Multimodal	several local minima	n	$x^* = (420.96, \ldots, 420.96)$, $f(x^*) = 0$

Efficiency: The efficiency refers to the computational effort required by the technique for constructing the meta-model and predicts the response for a new input. The efficiency of each meta-modeling technique is measured by the time used for its construction and the predictions performed with it. For these criteria, we decided to measure the period of time it takes to train a model with 100 solutions and perform the prediction of other 1000 solutions.

Suitability: These criteria refer to the degree of difficulty to optimize a meta-model created for a technique. It should be obvious that an accurate model can be more difficult to optimize, because their fitness landscape may be more rugged. To measure this criterion, we decided to use two approaches. First, we use a Differential Evolution algorithm in order optimize the surrogate model created by the approach at hand, after the application of the evolutionary approach, we will present the distance from the best solution obtained by the DE algorithm to the optima of the real test function. This process will be repeated 31 times for all the problem sizes in every test problem.

The second approach refers to the way of evolutionary algorithms compare solutions. They usually compare two solutions and select the one with the best fitness. Therefore, it would be interesting to build a meta-model, produce N distributed points with the meta-model at hand, and compare each pair of the generated points using both, the surrogate function and the real function, then, we will say that a meta-model approach is better than other, if it preserves better the comparative relation with respect to the real function. In order to measure this, we measure the degree of ranking preservation:

The ranking preservation (RP) refers to the ability of the meta-model to maintain the same rank of the solutions with respect to the original function. A meta-model $f'(x)$ has a perfect ranking preservation under the original function $f(x)$ if:

$$\begin{aligned} \forall x, y \in \mathcal{F} \quad &: \quad (f(x) < f(y) \Rightarrow f'(x) < f'(y)) \\ &\vee \quad (f(x) = f(y) \Rightarrow f'(x) = f'(y)) \end{aligned}$$

where $\mathcal{F}$ is the feasible region of the problem. Therefore, the metric can be defined as follows:

$$RP = \left(\sum_{i=1}^{N} \sum_{j=i+1}^{N} h(i,j) \right) / \binom{N}{2} \tag{5}$$

where $h(i,j)$:

$$h(i,j) = \begin{cases} 1 & \textbf{if}((f(i) < f(j) \wedge f'(i) < f'(j)) \\ & \vee (f(i) > f(j) \wedge f'(i) > f'(j)) \\ & \vee (f(i) = f(j) \wedge f'(i) = f'(j))) \\ 0 & \textbf{otherwise} \end{cases} \tag{6}$$

where N refer to the number of solutions used to validate the model. In this document, we used $N = 1000$ solutions for each size in every test problem.

For this performance measure, we will prefer a larger value of RP. In this case, since the value is normalized to $[0, 1]$, we will prefer those solutions closer to 1.

IV. COMPARISON OF RESULTS

A. Accuracy, robustness and scalability

We decided to used boxplots to present the results, since they show different descriptive measures simultaneously. The median is shown with a straight line inside the box. The average accuracy of the technique is shown with a diamond. The size of the box indicates the variability of the technique. Then, the smaller the box, the more robust the technique is.

Figure 1 shows the results produced by the application of the G-metric to the solutions obtained by each meta-model using two sets of parameter's configurations, 1) the best local settings (BLS) and 2) best overall settings (BOS) for the unimodal test functions (see Figure 1(a)), the multimodal test functions (see Figure 1(b)), and for the six test functions (see Figure 1(c)).

It is clear from this figures, that the algorithms behaved alike when optimizing both, unimodal and multimodal test functions. However, it is also clear that the studied approaches behaved better for the unimodal ones. We also found that both, BLS and BOS produced a similar behavior on each algorithm, except for the PRS approach, where, BLS presented a more consistent behavior than BOS. Then, we can say that the parameters previously found can be used to avoid a future tuning. Needless to say, these parameters are the ones who had

a better performance in general, but they were not the best for every problem, so that whenever possible a priori adjustment is better to carry it out.

Despite the graphics are not conclusive for KRG, RBF and SVR, it is possible to address some conclusions for the robustness behavior. If we analyze the size and position of the box, we can find that RBF behaved slightly better than KRG and both outperformed SVR. The worst performance in both performance measures (G-metric and robustness) was for PRS.

In order to improve the conclusions of this comparison, we decided to investigate the performance of the studied algorithms in each test function. In this experiment, it is more notorious the lack of accuracy of PRS, even for a low quantity of variables. We can see for Step and Ackley functions shown in Figures 2(a), 2(d), respectively, that PRS found solutions up to eight variables, but showing a poor accuracy. For Sphere, Rosenbrock and Rastrigin test functions, shown in Figures 2(b), 2(c) and 2(e), respectively, it could approximate for $v = [2, 4, 8, 10]$ problem sizes. However, again, the performance of the approach was the poorest. This approach could behave well only for Schwefel test function shown in Figure 2(f), where it could approximate the problem in all the tested sizes, and it showed the best performance for the 50 variables instance.

The SVR approach had an erratic behavior in Step and Ackley test functions for $v = [2, 4, 6, 8, 10]$ problem sizes (shown in Figures 2(a) and 2(d)), respectively, but performed considerable better than PRS and it increased its behavior for the rest of the test functions, sharing the same behavior with KRG for Sphere (see Figure 2(b)), Rastrigin (see Figure 2(e)) and Schwefel (see Figure 2(f), and only behind from it in $v = [2, 4, 6, 8, 10, 15]$ instances in Rosenbrock test function shown in Figure 2(c). KRG in the other hand, had an impressive performance for small sizes in all the problems ($v = [2, 4, 6, 8, 10, 15]$), outperforming the other techniques in these problem sizes. However, its performance decreases after 15 variables in all test functions as can be seen in Figure 2. This reduction in the performance is well justified since it is natural that an increase in the number of variable lead to an increase in prediction error.

Contrary to the KRG behavior, RBF had a poor performance with the initial sizes of all problems as is easily seen in Figure 2, but it was capable to maintain its performance in bigger sizes of all problems. Although we believe it is possible to improve more its accuracy with a better selection of centers.

Figure 3 shows the performance of the techniques in the six test functions. KRG was the technique that showed the better performance for a few variables ($v < 15$), and RBF was the technique with better performance for high-dimensional functions. The technique with the worst overall performance was PRS. SVR achieved similar performance than KRG, although this was slightly better. RBF is very robust with respect to the increment of the number of variables for all problems, since their results were held constant.

We associated the size of the problem with respect to the dimensionality, the high dimensionality for problems with $v > 15$ and low dimensionality for others. In Table III show that KRG was the best approach for low dimensionality problems. Moreover, RBF is the best for high dimensionality problems. Overall, RBF was the best technique with respect to the accuracy and the scalability.

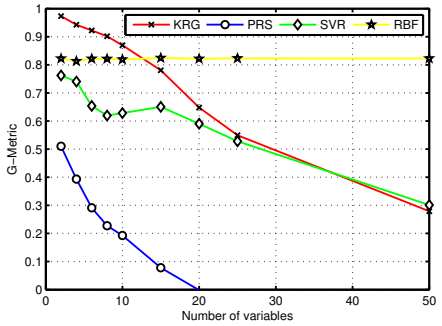


Fig. 3. The mean for accuracy metric by number of variables in all the problems

TABLE III
BEST TECHNIQUES WITH RESPECT TO THE ACCURACY AND DIMENSIONALITY: ALL THE PROBLEMS

	Unimodal	Multimodal	Overall
Low dimensionality	KRG	KRG	KRG
High dimensionality	RBF	RBF	RBF
Overall	RBF	RBF	RBF

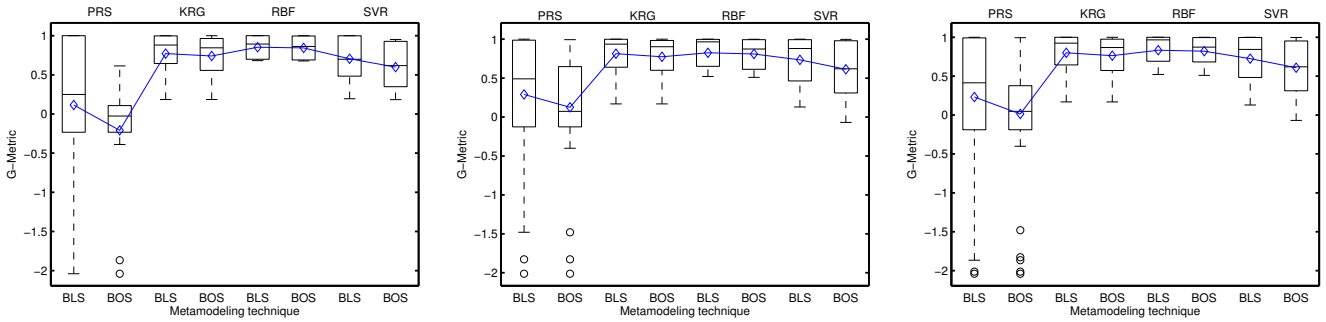
As a main feature of the results observed we can see that exist a point where the performances for KRG and RBF intersects, this point of intersection can be used as reference of when to use a meta-model or another.

B. *Efficiency*

As mentioned in Section III, the efficiency was measured with respect to the time it takes for each technique to train a model with 100 solutions and predict 100 new ones. The results for this metric are shown in Figure 4. Contrary to what was expected, PRS was not the technique that required less time, probably because of the technique used for constructing the polynomial (Stepwise). It is found that, for KRG, the time consuming is relatively large, mainly caused the embedded optimization method to find the best values of the parameters θ. Moreover, in SVR and RBF the time remained constant.

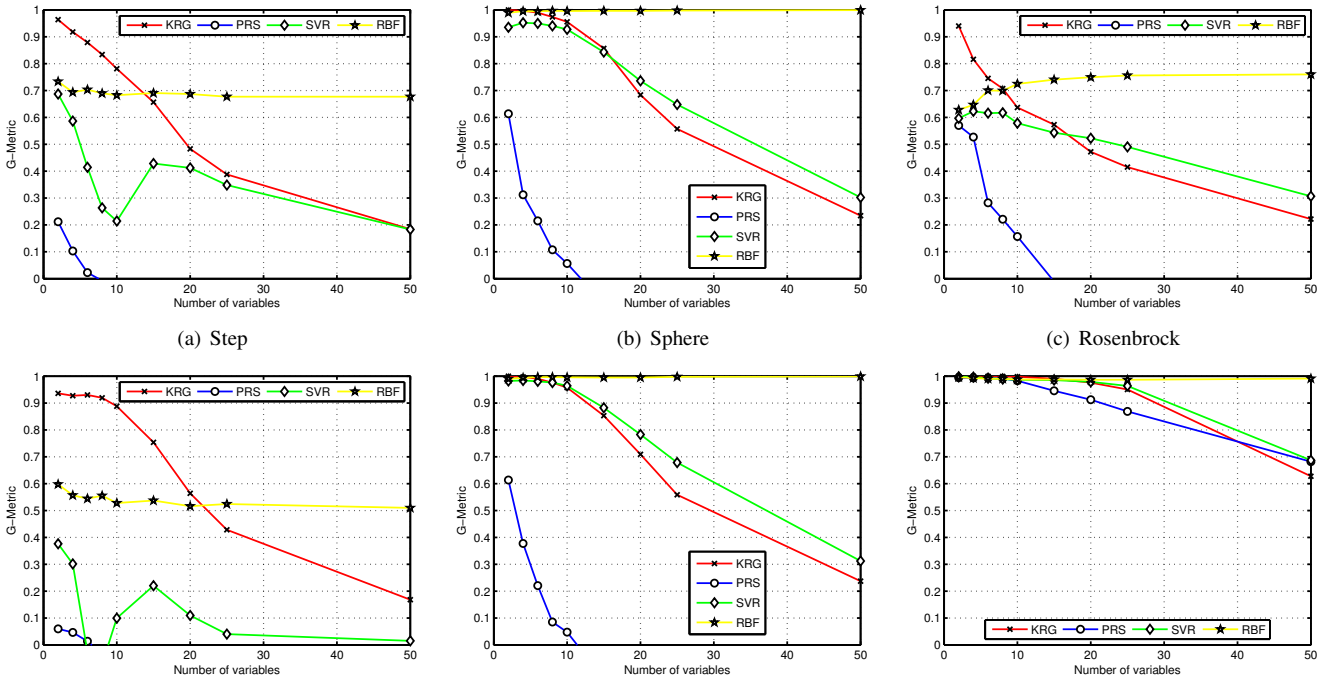
C. *Suitability*

It is natural to expect that if we approximate a rugged objective function using a surrogate model, then, the fitness landscape created by the artificial function will also be rugged, and therefore, it will be difficult to be optimized. As it was mentioned above, in order to measure the ability of an evolutionary algorithm to optimize the fitness landscape produced by a specific surrogate model, we decided to optimize the models produced by the different surrogate techniques with



(a) Unimodal problems (b) Multimodal problems (c) All the problems

Fig. 1. Accuracy for the meta-modeling techniques.



(a) Step (b) Sphere (c) Rosenbrock

(d) Ackley (e) Rastrigin (f) Schwefel

Fig. 2. Average accuracy by number of variables

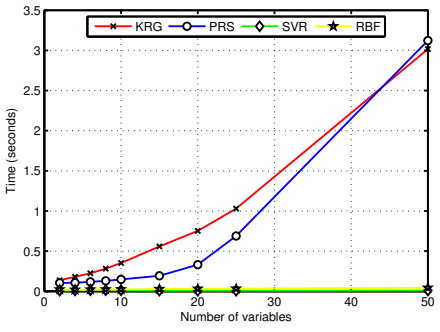


Fig. 4. Average time of exectution for training+prediction in all the problems

a Differential Evolution algorithm. However, as a drawback of this experiment is that, the associated optimization will be also attached to the accuracy of the meta-model itself. Since the optima solution on each real test function is well-known, then, we will say that a meta-model approach A is easier to optimize than another B, if the result from the best solution found in a fixed number of iterations by the approach A is closer to the optima than the one from the B approach.

The results from this experiment are shown in Figure 5. It can be seen that the results are closely linked with the accuracy of the meta-model, because most models were easier to optimize. From this experiment, we can conclude that RBF was the best approach to be used in an evolutionary approach. The results obtained for KRG and SVR were very similar. Therefore, we can use them indistinctly. Finally, PRS was the approach with the worst performance, probably the poor associated performance was due to the low accuracy of the meta-model. Moreover, it is easily to see that a raised in

the number of variables also increased the complexity of the search in the evolutionary approach.

Since we decided to use a second performance metric to measure the suitability, we selected the RP performance measure. The methodology used consisted on train each meta-model with a dataset of size 100, and then, measure the RP with $N = 1000$ solutions. The procedure was applied to the six test problems with all instance's sizes. From the results shown in Figure 6, we can say that the approach with best RP was RBF. Moreover, clearly SVR was the technique with the worst RP in general for all the problems. Figure 7 shows the results of the RP metric. From this figure, we can see that RBF is very consistent in all problem sizes. Although this study only shows the overall results, these indicate that PRS had a good performance up to $v = 20$ in most of the problems with respect to SVR; and KRG presented the worse performance with respect to the raise of the dimensionality.

An important feature of RP is that the higher is its value, the lower the probability of add a false optimum to the problem. This feature is a very important characteristic when optimizing with evolutionary algorithms.

With the analysis of the results, it can be seen that if the meta-model had a poor performance in accuracy and a good performance for RP, we can expect a good behavior in the optimization process. However, if the meta-model had a good performance in accuracy but a bad performance for RP, the behavior not necessarily would be bad in the optimization process. The above justified the good performance of the Differential Evolution with the RBF meta-model.

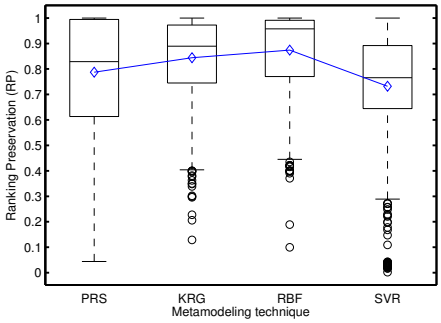


Fig. 6. Ranking preservation in all the problems with all the problem sizes for each meta-model

V. Conclusions

The purpose of the current study was to compare meta-modeling techniques and to evaluate them under multiple aspects in order identify their suitability to be used with evolutionary algorithms. The study presented in this paper has provided results about the performance of PRS, KRG, RBF and SVR techniques in several test problems with different features.

From the results derived by this study, we can say that the best approach to be used in low dimensionality problems, can be KRG or even SVR. In contrast if we try to optimize a high dimensional problem, then, the best technique to be used would be RBF. Moreover, if the evolutionary algorithm will be used as an algorithm to solve different kinds of problems, probably you need to use a more robust meta-modeling technique. Therefore, we will suggest, again, the use of RBF.

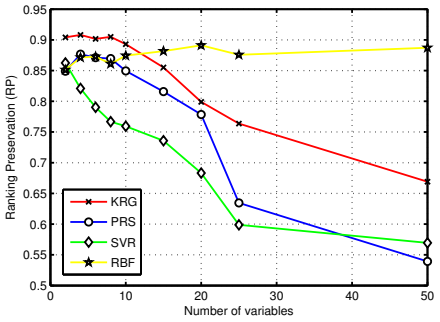


Fig. 7. Ranking preservation in all the problems by number of variables

In addition, it should be noted that not necessarily the less accurate meta-model will be the one with the worst results in the optimization process. Imagine the worst case where the meta-model is totally inaccurate, but it also presents the best case in the RP metric, so, we will say, that the fitness landscape will be shifted, but it will be continued having the same shape.

With respect to efficiency, it is obvious that RBF and SVR are the winners in this aspect. However, in the most of the Evolutionary Algorithms we need the retraining of the meta-model several times, leading to a loss of performance. However, we propose in this work a set of parameters that behaved comparable to the best set of parameters found.

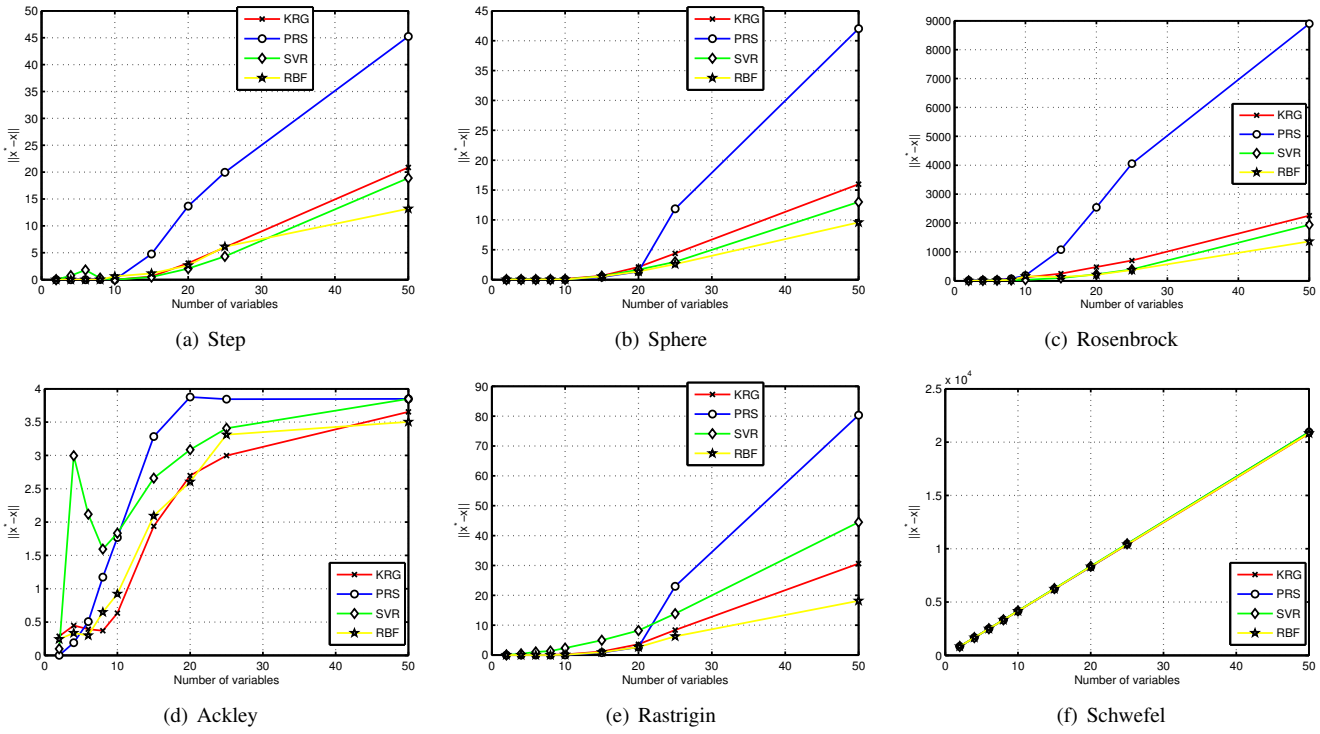
Further investigation and experimentation into the fitness landscape of the meta-models and the use of multiple surrogates for approximate all the fitness landscape is strongly recommended. Likewise, it would be interesting to assess other aspects such as simplicity of the surrogate model or measure the efficiency with respect to another factor different to the time.

Acknowledgment

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Information Technology Laboratory, CINVESTAV-Tamaulipas. The second author gratefully acknowledges support from CONACyT through project 105060. Also, this research was partially funded by project number 51623 from "Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas". Finally, we would like to thank to "Fondo Mixto de Fomento a la Investigación científica y Tecnológica CONACyT - Gobierno del Estado de Tamaulipas" for their support to publish this paper.

References

[1] A. Ratle, "Optimal Sampling Strategies for Learning a Fitness Model," in *Proceedings of 1999 Congress on Evolutionary Computation*, vol. 3, Washington D.C., July 1999, pp. 2078–2085.



(a) Step (b) Sphere (c) Rosenbrock

(d) Ackley (e) Rastrigin (f) Schwefel

Fig. 5. Optimization of meta-modeling technique

[2] M. El-Beltagy and A. Keane, "Evolutionary Optimization for Computationally Expensive Problems using Gaussian Processes," in *Proceedings of International Conference on Artificial Intelligence*. CSREA, 2001, pp. 708–714.

[3] M. Emmerich, A. Giotis, M. Özdenir, T. Bäck, and K. Giannakoglou, "Metamodel-Assisted Evolution Strategies," in *Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, no. 2439. Springer, 2002, pp. 371–380.

[4] D. Bueche, N. Schraudolph, and P. Koumoutsakos, "Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models," *IEEE Trans. on Systems, Man, and Cybernetics: Part C*, 2004, in press.

[5] R. G. Regis and C. A. Shoemaker, "Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 490–505, 2004. [Online]. Available: http://dx.doi.org/10.1109/TEVC.2004.835247

[6] W. Carpenter and J. F. Barthelemy, "A Comparison of Polynomial Approximation and Artificial Neural Nets as Response Surface," AIAA, Tech. Rep. 92-2247, 1992.

[7] T. Simpson, T. Mauery, J. Korte, and F. Mistree, "Comparison of Response Surface and Kriging Models for Multidiscilinary Design Optimization," AIAA, Tech. Rep. 98-4755, 1998.

[8] A. A. Giunta and L. T. Watson, "A Comparison Of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," 1998.

[9] L. Willmes, T. Baeck, Y. Jin, and B. Sendhoff, "Comparing Neural Networks and Kriging for Fitness Approximation in Evolutionary Optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2003, pp. 663–670.

[10] R. Jin, W. Chen, and T. Simpson, "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria," AIAA, Tech. Rep. 2000-4801, 2000.

[11] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[12] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–423, November 1989. [Online]. Available: http://dx.doi.org/10.1214/ss/1177012413

[13] R. L. Hardy, "Multiquadric Equations of Topography and Other Irregular Surfaces," *J. Geophys. Res.*, vol. 76, pp. 1905–1915, 1971.

[14] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, September 1998. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0471030031

[15] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, University of Michigan, Ann Arbor, MI, USA, 1975, aAI7609381.

[16] H. Mühlenbein, M. Schomisch, and J. Born, "The Parallel Genetic Algorithm as Function Optimizer," *Parallel Computing*, vol. 17, no. 6-7, pp. 619 – 632, 1991.

[17] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[18] H. Schwefel, *Numerical Optimization of Computer Models*. New York, NY, USA: John Wiley & Sons, Inc., 1981.

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