

A Study on Metamodeling Techniques, Ensembles, and Multi-Surrogates in Evolutionary Computation

Dudy Lim
School of Computer
Engineering
Nanyang Technological
University, Nanyang Avenue,
Singapore 639798
dlim@ntu.edu.sg

Yew-Soon Ong
School of Computer
Engineering
Nanyang Technological
University, Nanyang Avenue,
Singapore 639798
asysong@ntu.edu.sg

Yaochu Jin
Honda Research Institute
Europe GmbH
Carl-Legien Strasse 30,
Offenbach 63073
yaochu.jin@honda-ri.de

Bernhard Sendhoff
Honda Research Institute
Europe GmbH
Carl-Legien Strasse 30,
Offenbach 63073
bernhard.sendhoff@honda-ri.de

ABSTRACT

Surrogate-Assisted Memetic Algorithm(SAMA) is a hybrid evolutionary algorithm, particularly a memetic algorithm that employs surrogate models in the optimization search. Since most of the objective function evaluations in SAMA are approximated, the search performance of SAMA is likely to be affected by the characteristics of the models used. In this paper, we study the search performance of using different metamodeling techniques, ensembles, and multi-surrogates in SAMA. In particular, we consider the SAMA-TRF, a SAMA model management framework that incorporates a trust region scheme for interleaving use of exact objective function with computationally cheap local metamodels during local searches. Four different metamodels, namely Gaussian Process (GP), Radial Basis Function (RBF), Polynomial Regression (PR), and Extreme Learning Machine (ELM) neural network are used in the study. Empirical results obtained show that while some metamodeling techniques perform best on particular benchmark problems, ensemble of metamodels and multi-surrogates yield robust and improved solution quality on the benchmark problems in general, for the same computational budget.

Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—*global optimization*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'07, July 7-11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

General Terms

Algorithm

Keywords

Surrogate-assisted Memetic Algorithm, Evolutionary Algorithm using Approximation

1. INTRODUCTION

Evolutionary Algorithm(EA), a stochastic computational model inspired by the neo-Darwinian theory of evolution, has been used as the major optimization framework in various complex real-world optimization problems, such as in the designs of aerodynamic shape [1], transonic civil transport aircraft wing [2], stator blade [3], hard disk drive servo control [4], and biomedical applications [5]. However, as a population-based algorithm, thousands of calls to the analysis codes are often required to locate a near optimal solution in most conventional EA. The rising trend of using time-consuming simulation codes in scientific and engineering works has further increased the impracticality of EA as a global optimization tool. To alleviate this problem, in EA it has been a standard practice for computationally cheap approximation or surrogate models to be used in lieu of exact model, hence the term Surrogate-Assisted Evolutionary Algorithm (SAEA). Among these techniques, Polynomial Regression (PR, also known as response surface method), Artificial Neural Network (ANN), Radial Basis Function (RBF), and Gaussian Process (GP) (also referred to as Kriging or Design and Analysis of Computer Experiments (DACE) models) are the most prominent and commonly used [6][7]. In [6], Ratle proposed a strategy for integrating GA with Kriging metamodel and uses a heuristic convergence criterion to decide when the model should be updated. The work was extended by El-Beltagy et al. [8] who considered the issue of balancing the concerns of optimization with those of Design of Experiments (DOE). Jin et al. proposed the coupling of ES with neural network mod-

els in [9]. The concept of generation control and individual control in the evolutionary search was introduced. Further, some empirical criteria for switching between the exact fitness function and approximate models throughout the EA search are provided. Other ideas on using pre-selection to decide the portions of the EA population that undergoes exact fitness evaluations were also considered in [10].

In [11], Ong et. al. propose a surrogate assisted memetic algorithm for solving optimization problems with computationally expensive fitness function and general constraints, on a limited computational budget. The essential backbone of the framework is an evolutionary algorithm coupled with a trust region managed feasible sequential quadratic programming solver in the spirit of Lamarckian learning. The TRF is used for interleaving use of exact models for the objective and constraint functions with computationally cheap surrogate models during local search. Extensions to enhance search efficiency and approximation accuracy using gradient information and multi-level surrogates were also considered in [13] and [14] recently. In this paper, we perform a detail study on the search performance of SAMA when different approximation metamodels are employed. It is common knowledge that different metamodels have heterogeneous characteristics, hence the SAMA, which is heavily dependent on approximation of the fitness functions, is likely to be affected by the modeling methods used. We investigate these effects in SAMA-TRF, which was introduced in [11]. In total, four different metamodels, namely Gaussian Process (GP), Radial Basis Function (RBF), Polynomial Regression (PR), and Extreme Learning Machine (ELM) Neural Network are investigated in this study.

To facilitate our discussion, the remainder of this paper is structured as follows. In Section 2, we review the standard Surrogate-Assisted Memetic Algorithm (SAMA) and its variant, SAMA-TRF, which is used throughout this paper. The four different metamodels applied in the study are briefly elaborated in Section 3. Section 4 then presents an empirical study on different combination of metamodels and benchmark functions. In this section, a study on using multiple metamodels in SAMA is also presented, both in the forms of an ensemble of metamodels and multi-surrogates. Finally, section 5 concludes this paper.

2. SURROGATE ASSISTED MEMETIC ALGORITHMS

Memetic Algorithms (MAs) are population-based meta-heuristic search methods that are inspired by Darwinian principles of natural evolution and Dawkins notion of a meme defined as a unit of cultural evolution that is capable of local refinements.

A unique property of MA is the heavy use of the local search strategy throughout the entire evolutionary search. If designed correctly, they should converge to high quality solutions more efficiently than conventional evolutionary algorithms [15].

Note that no form of approximations is employed in the standard MA. Hence, the core idea of the SAMA is to reduce the number of calls to the computationally expensive function by replacing the exact fitness function used in the local search phase of the standard MA with computationally cheap surrogates. The SAMA-TRF incorporates a Trust-Region Framework (TRF) in the standard SAMA for inter-

BEGIN

Initialize: Generate a database containing a population of designs.
While(*computational budget is not exhausted*)

- Evaluate all individuals in the population using the exact fitness function.
- **For** each non-duplicated individual in the EA population
 - **MA:** Apply local search strategy using original exact fitness functions.
 - **SAMA:** Apply local search strategy using online surrogates.
 - **SAMA-TRF:** Apply k iterations of TRF-regulated local search strategy using online surrogates.

End For

- Replace the individuals in the population with the locally improved solution in the spirit of Lamarckian learning.
- Apply standard EA operators to create a new population.

End While
END

Figure 1: Pseudo codes of the standard MA, SAMA, and SAMA-TRF

leaving use of exact objective function with computationally cheap local surrogate models during local searches [16]. Since gradient-based local search algorithm involved in EAs makes use of line searches to locate a new iterate, the issue of range of validity of the surrogate models or the control of approximation errors can be directly addressed by using *ad hoc* move limits or a trust region framework. As shown by Alexandrov *et al.* [17], the trust-region strategy for adaptively controlling the move limits guarantees global convergence under some mild assumptions on the accuracy of the surrogate model. The core differences between the standard MA, SAMA, and SAMA-TRF are highlighted in Fig. 1.

The Trust Region Framework (TRF) is used to ensure the convergence to the local optimum of the exact computationally expensive fitness function [16], [12]. More specifically, for each non-duplicated individuals in the population, the local search proceeds with a sequence of trust-region subproblems of the form

$$\text{Minimize : } \hat{f}^k(\mathbf{x}_c^k + \Omega^k) \quad (1)$$

where $k = 0, 1, 2, \dots, k_{max}$, $\hat{f}(x)$ is the approximation function corresponding to the objective function $f(x)$, \mathbf{x}_c^k and Ω^k are the initial guess and the trust-region radius used for local search at iteration k , respectively.

For each subproblem (or during each trust-region iteration), surrogate models of the exact fitness function, viz., $\hat{f}^k(\mathbf{x})$ is created dynamically. The m nearest neighbors of the initial guess, \mathbf{x}_c^k , are extracted from the archived database of design points evaluated so far using the exact analysis codes. These points are then used to construct local surrogate models of the exact objective function.

The surrogate models thus created are used to facilitate the necessary fitness function estimations in the local searches. During local search, we initialize the trust-region Ω using the minimum and maximum values of the design points used to construct the surrogate model. After each iteration, the trust-region radius Ω^k is updated based on a measure which

indicates the accuracy of the surrogate model at the k th local optimum, \mathbf{x}_{lo}^k . After computing the exact values of the fitness function at this point, the figure of merit, ρ^k , is calculated as

$$\rho^k = \frac{f(\mathbf{x}_c^k) - f(\mathbf{x}_{lo}^k)}{\hat{f}(\mathbf{x}_c^k) - \hat{f}(\mathbf{x}_{lo}^k)} \quad (2)$$

The above equations provide a measure of the actual versus predicted change in the exact fitness function values at the k th local optimum. The value of ρ^k is then used to update the trust-region radius as follows [17]:

$$\begin{aligned} \Omega^{k+1} &= 0.25\Omega^k, & \text{if } \rho^k \leq 0.25, \\ &= \Omega^k, & \text{if } 0.25 < \rho^k \leq 0.75, \\ &= \xi\Omega^k, & \text{if } \rho^k > 0.75, \end{aligned} \quad (3)$$

where $\xi = 2$, if $\|\mathbf{x}_{lo}^k - \mathbf{x}_c^k\|_\infty = \Omega^k$ or $\xi = 1$, if $\|\mathbf{x}_{lo}^k - \mathbf{x}_c^k\|_\infty < \Omega^k$.

The trust-region radius, Ω^k , is reduced if the accuracy of the surrogate, measured by ρ^k is low. Ω^k is doubled if the surrogate is found to be accurate and the k th local optimum, \mathbf{x}_{lo}^k , lies on the trust-region bounds. Otherwise the trust-region radius remains unchanged.

The exact solutions of the fitness functions at the k th local optimum are combined with the existing neighboring data points to generate new surrogate models in the subsequent trust-region iterations. The initial guess for the $k+1$ iteration is defined by

$$\begin{aligned} \mathbf{x}_c^{k+1} &= \mathbf{x}_{lo}^k, & \text{if } \rho^k > 0 \\ &= \mathbf{x}_c^k, & \text{if } \rho^k \leq 0. \end{aligned} \quad (4)$$

The trust-region process for an individual terminates when the maximum number of trust-region iterations permissible, k_{max} , and configurable by the user is reached. Lamarckian learning then proceeds if the k_{max} local optimum solution obtained is an improvement over that of the initial individual.

3. METAMODELING TECHNIQUES

In this section, we briefly discuss four different meta-models used for comparison in this paper, i.e. Gaussian Process(GP), Radial Basis Function(RBF), Polynomial Regression(PR), and Extreme Learning Machine(ELM) Neural Network.

Throughout the discussion, we assume the following. Let $\mathcal{D} = \{\mathbf{x}_i, t_i\}, i = 1 \dots n$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is the input design vector, $t_i \in \mathbb{R}$ is the corresponding target value, and $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})$, d denotes the dimension of the problem, we have $t_i = f(\mathbf{x}_i) = f(x_{i1}, \dots, x_{id})$.

3.1 Kriging / Gaussian Process (GP)

The GP surrogate model [19] assumes the presence of an unknown true modeling function $f(\mathbf{x})$ and an additive noise term v to account for anomalies in the observed data. Thus:

$$t = f(\mathbf{x}) + v \quad (5)$$

From a stochastic process viewpoint, the collection $\mathbf{t} = \{t_1, t_2, \dots, t_n\}$ is called a Gaussian process if every subset of \mathbf{t} has a joint Gaussian distribution. More specifically,

$$P(\mathbf{t}|\mathbf{C}, \{x_n\}) = \frac{1}{Z} \exp\left(-\frac{1}{2}(\mathbf{t} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{t} - \boldsymbol{\mu})\right) \quad (6)$$

where \mathbf{C} is a covariance matrix parameterized in terms of hyperparameters $\boldsymbol{\theta}$, i.e., $\mathbf{C}_{ij} = k(x_i, x_j; \boldsymbol{\theta})$ and $\boldsymbol{\mu}$ is the process mean. The Gaussian process is characterized by this covariance structure since it incorporates prior beliefs both about the true underlying function as well as the noise model. In the present study, we use the following exponential covariance model

$$k(x_i, x_j) = e^{-(x_i - x_j)^T \Theta (x_i - x_j)} + \theta_{d+1} \quad (7)$$

where $\Theta = \text{diag}\{\theta_1, \theta_2, \dots, \theta_d\} \in \mathbb{R}^{d \times d}$ is a diagonal matrix of undetermined hyperparameters, and $\theta_{d+1} \in \mathbb{R}$ is an additional hyperparameter arising from the assumption that noise in the dataset is Gaussian (and output dependent). We shall henceforth use the symbol $\boldsymbol{\theta}$ to denote the vector of undetermined hyperparameters, i.e., $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_{d+1}\}$. In practice, the undetermined hyperparameters are tuned to the data using the evidence maximization framework.

3.2 Radial Basis Function (RBF)

In RBF, the local surrogate models are interpolating radial basis function networks of the form

$$\hat{t} = \hat{f}(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\|\mathbf{x} - \mathbf{x}_i\|) \quad (8)$$

where $K(\|\mathbf{x} - \mathbf{x}_i\|) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a RBF and $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n\} \in \mathbb{R}^n$ denotes the vector of weights. Typical choices for the kernel include linear splines, cubic splines, multiquadrics, thin-plate splines, and Gaussian functions [20].

3.3 Polynomial Regression (PR)

In PR model [21], we define an exponent vector $\boldsymbol{\varepsilon}$ containing positive integers $(\pi_1, \pi_2, \dots, \pi_d)$ and define $\mathbf{x}_i^\boldsymbol{\varepsilon}$ as an exponent input vector $(x_{i1}^{\pi_1}, x_{i2}^{\pi_2}, \dots, x_{id}^{\pi_d})$.

Given a set of exponent vectors $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_m$ and the set of data (\mathbf{x}_i, t_i) , where $i = 1, 2, \dots, n$, the polynomial model of $(m-1)^{th}$ order has the form:

$$\hat{t}_i = C_1 \mathbf{x}_i^{\boldsymbol{\varepsilon}_1} + C_2 \mathbf{x}_i^{\boldsymbol{\varepsilon}_2} + \dots + C_m \mathbf{x}_i^{\boldsymbol{\varepsilon}_m} \quad (9)$$

where C_1, C_2, \dots, C_m are the coefficient vectors to be estimated, and $C_j = (c_{j1}, c_{j2}, \dots, c_{jd})$, $j = 1, 2, \dots, m$. The least square method is then used to estimate the coefficients of the polynomial model.

3.4 Extreme Learning Machine (ELM) s Neural Network

The ELM Neural Network [22] consists of three layers: input, hidden, and output layers. The input weights are assigned randomly while the output weights are determined analytically using Moore-Penrose pseudoinverse formulated as follows:

$$W = \left(H^T H\right)^{-1} H^T t \quad (10)$$

where $W = (w_1, w_2, \dots, w_d)$ is the output weights, H is an $h \times n$ matrix as the output of an h -neuron hidden layer for n data, and $t = (t_1, t_2, \dots, t_n)$ denotes the final prediction value at the output layer. Since no iterative computations are needed, ELM trains very fast when compared to technique such as Backpropagation algorithm [22][23].

4. EMPIRICAL STUDY

In this section, we present an empirical study on SAMA-TRF using the four approximation methods discussed in previous section. Several commonly used benchmark test functions in evolutionary optimization, i.e., Ackley, Griewank, Rosenbrock, and Step functions, used in the present study, are formulated in equations 11, 12, 13, and 14, respectively. Without loss of generality, in this study, 10D functions are considered. Please note that for simplicity, in this section the term SAMA and its variant, SAMA-TRF are used interchangeably as only SAMA-TRF is considered.

Ackley

$$f(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} \quad (11)$$

$$-32.768 \leq x_i \leq 32.768, i = 1, 2, \dots, n.$$

Griewank

$$f(x) = 1 + \sum_{i=1}^n x_i^2/4000 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) \quad (12)$$

$$-600 \leq x_i \leq 600, i = 1, 2, \dots, n.$$

Rosenbrock

$$f(x) = \sum_{i=1}^{n-1} (100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \quad (13)$$

$$-2.048 \leq x_i \leq 2.048, i = 1, 2, \dots, n - 1.$$

Step

$$f(x) = \sum_{i=1}^n [x_i^2] \quad (14)$$

$$-5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, n.$$

For the metamodels used, the following settings are applied. Linear spline is used as the kernel function for the RBF model. The PR model is a second order model, while the ELM Neural Network uses 100 hidden neurons and unipolar sigmoid activation function. For the GP model, the hyperparameters are determined using evidence maximization framework as discussed in previous section.

With the maximum computational budget allowable configured as 6000 exact fitness evaluations, search trends of Simple GA, MA, and SAMA with different metamodels are plotted in Figures 2-5 which are averaged over 20 independent runs. All optimization schemes used are configured to have population size of 50, uniform crossover and mutation with probability of 0.9 and 0.01, respectively. Elitism and ranking selection are also used in the experiments. The trust region iteration used, k , for the SAMA is configured to be 3, as suggested in [11]. Using $k=3$, a single local search on each individual will consume three exact fitness evaluations. The number of nearest neighbours used to build local approximation model, m is configured to be 10 times the problem dimensionality, i.e. 100 points for the 10D problems. It is worth noting that all these settings are determined empirically from some preliminary runs. To obtain a database to be used by the models in early stages of the optimization, the first 10 generations uses the Simple GA, hence the first 500 exact fitness evaluations result in similar convergence trends as can be observed in the figures.

From these results, it is notable that most of the SAMA variants considered here, are capable of converging to good solution quality more efficiently than standard GA and MA on the benchmark problems. This makes good sense since SAMA, which employs local search heavily using approximation, generally search more efficiently. Most importantly,

since the surrogates are used in place of the exact fitness function when conducting local searches, the SAMA-RBF, SAMA-GP, SAMA-PR, and SAMA-ELM incurs significantly lower computational efforts than the standard GA for the same search generations. However, on the unimodal Rosenbrock function, standard MA performs competitively to the SAMA variants.

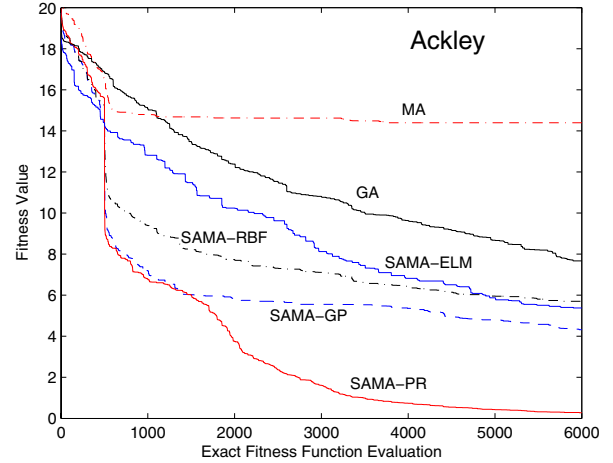


Figure 2: Convergence trends of Simple GA, MA, SAMA-RBF, SAMA-GP, SAMA-PR, and SAMA-ELM for Ackley function.

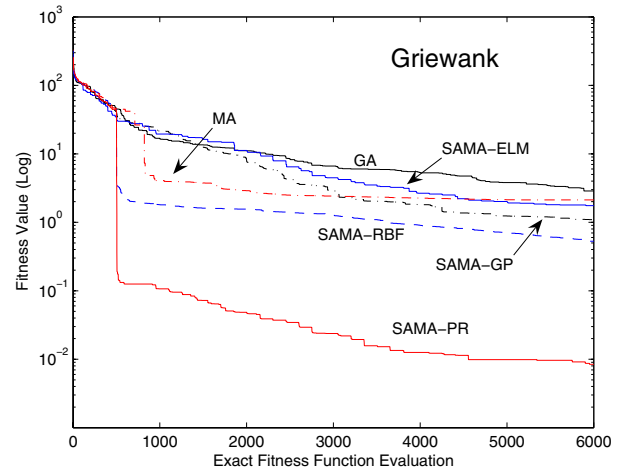


Figure 3: Convergence trends of Simple GA, MA, SAMA-RBF, SAMA-GP, SAMA-PR, and SAMA-ELM for Griewank function.

4.1 Quality Measures of Metamodels in SAMA

In this subsection, we investigate the quality of the metamodels used and how they relate to the search performance of SAMA. To begin, we focus on the two criteria commonly used to assess the quality of surrogate models, i.e. the root mean square error ($rmse$) and correlation coefficient (r).

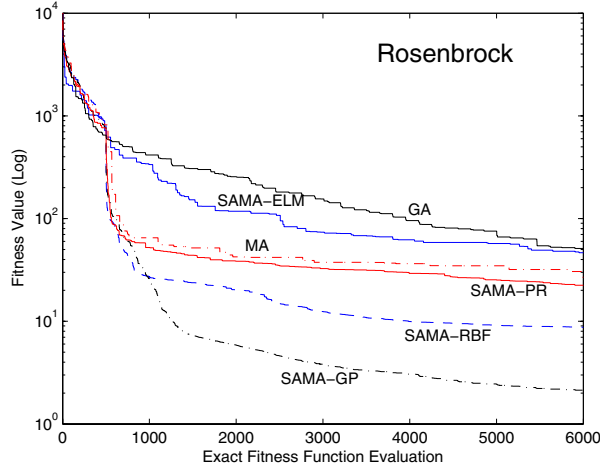


Figure 4: Convergence trends of Simple GA, MA, SAMA-RBF, SAMA-GP, SAMA-PR, and SAMA-ELM for Rosenbrock function.

If $f(\mathbf{x})$ denotes the original fitness function and the approximated function is $\hat{f}(\mathbf{x})$, the approximation errors at any design point \mathbf{x}_i is $e(\mathbf{x}_i)$, i.e., the uncertainty introduced by the surrogate at \mathbf{x}_i , may then be defined as

$$e(\mathbf{x}_i) = |f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)| \quad (15)$$

For each non-duplicated individual, if n fitness calls to $\hat{f}(\mathbf{x})$ are made in the SAMA local search strategy, the root mean square error, $rmse$ can be derived as

$$rmse = \sqrt{\frac{\sum_{i=1}^n e^2(\mathbf{x}_i)}{n}}. \quad (16)$$

While the correlation coefficient r is defined by

$$r = \frac{N \sum t \hat{t} - \sum t \sum \hat{t}}{\sqrt{[N \sum t^2 - (\sum t)^2][N \sum \hat{t}^2 - (\sum \hat{t})^2]}}. \quad (17)$$

where N is the sample size used. t denotes the exact fitness values from the independent test set, and \hat{t} are the fitness values estimated by the surrogate. If a correlation coefficient of 1 is obtained, this implies that the surrogate models the test set accurately.

Locally optimized solutions using the different models in the SAMA run are archived to form the sample set for evaluating the quality of different surrogates. The obtained $rmse$ and r measures when used in optimizing the benchmark functions are reported in Tables 1 and 2, respectively. The best performing metamodeling technique of each function is highlighted in *italic*.

Benchmark Functions	$rmse$			
	GP	PR	RBF	ELM
Ackley	1.135	3.127	<i>0.589</i>	19.832
Griewank	2.117	<i>0.250</i>	2.597	15.341
Rosenbrock	<i>25.473</i>	99.712	67.619	196.421
Step	<i>0.533</i>	1.418	0.864	12.309

Table 1: Criterion-1 ($rmse$) measures for GP, PR, RBF, and ELM surrogates

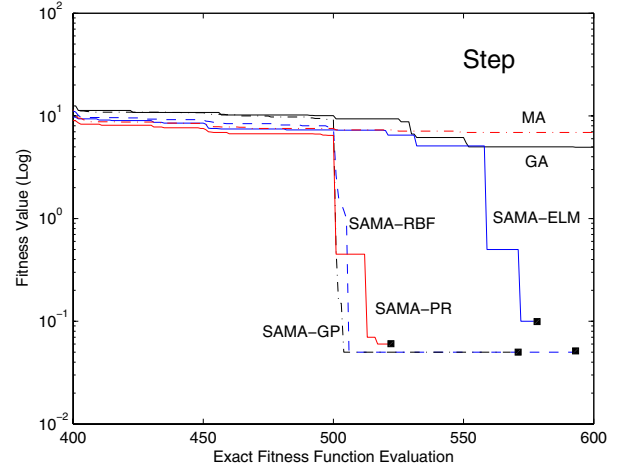


Figure 5: Convergence trends of Simple GA, MA, SAMA-RBF, SAMA-GP, SAMA-PR, and SAMA-ELM for Step function.

Benchmark Functions	r correlation			
	GP	PR	RBF	ELM
Ackley	0.964	0.834	<i>0.995</i>	0.542
Griewank	0.957	0.934	<i>0.979</i>	0.799
Rosenbrock	0.796	0.736	<i>0.829</i>	0.527
Step	0.901	0.947	<i>0.964</i>	0.412

Table 2: Criterion-2 (r correlation) measures for GP, PR, RBF, and ELM surrogates

Benchmark Functions	Rank of Search Performance			
	GP	PR	RBF	ELM
Ackley	2	<i>1</i>	4	3
Griewank	3	<i>1</i>	2	4
Rosenbrock	<i>1</i>	3	2	4
Step	3	<i>1</i>	2	4

Table 3: Criterion-3 (final solution quality) measures for GP, PR, RBF, and ELM surrogates

It is also worth noting that in optimization, it would be more interesting in predicting search improvement in the context of optimization as opposed just to the quality of the approximation. Hence, a third criterion is introduced here based on the rank of the problem solution quality for a fixed computational budget. The solution quality of the benchmark functions for different metamodels at the end of 6000 exact evaluations is summarized in Table 3. In terms of $rmse$, i.e. the measures of absolute errors in the prediction, GP exhibits the least error among all four. On the other hand, RBF appears best as far as the correlation measure, i.e. the intensity of association between the exact and predicted data set, is concerned. However, it is worth noting that the numerical values tabulated in Tables 1-3 do not reflect this straightforward relationship. Further, even though PR does not exhibit any good accuracy on both $rmse$ and r measures, it generates the best solution quality on 3 out of the 4 benchmark problems considered. Obviously, the generalizing capabilities of the 2nd-order PR leads to smoothing in the multimodal characteristics of the original(exact) fitness function, leading to faster convergence

rate. The same observations was reported in [24][26] and termed as the ‘*blessing of uncertainty*’ in [25]. Nevertheless, it is also worth noting that while the ‘*blessing of uncertainty*’ may be useful, to some extent, mild assumption on the accuracy of the metamodeling technique must be maintained. For instance, the ELM (with the configuration used in this paper) leads to poor solution quality in SAMA due to the significantly poor accuracies in both *rmse* and *r* measures (see Tables 1-3).

4.2 SAMA Using Ensemble of Metamodels and Multi-Surrogates

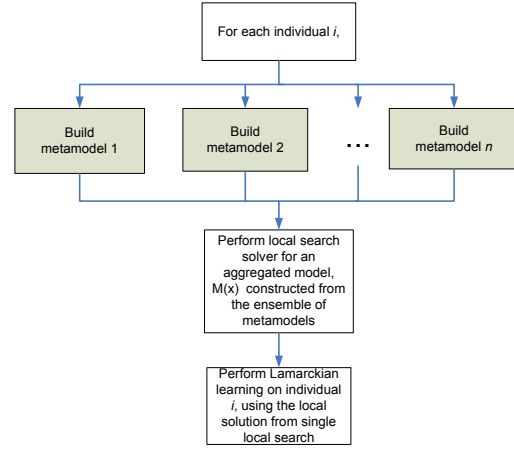
From the numerical results obtained in previous subsection, it is worth keeping in mind that uncertainty introduced by approximation errors in the surrogate model can benefit as well as obstruct effective optimization search. Since there is often no prior knowledge on which surrogates or ensembles would be most appropriate for the problem of interest, the use of multiple metamodels in the SAMA search appears to be an intuitive alternative. It is worth noting that the use of multiple metamodels can come in various forms. Recent studies in [27] and [28] show that using different metamodels in multiple local searches or in the form of an ensemble, respectively, generally leads to improved generalization property and solution quality, representing some promising approaches for mitigating the effects of ‘*curse of uncertainty*’ as well as benefiting from ‘*bles of uncertainty*’.

In this study, we consider an Ensemble-SAMA (ESAMA) and several variants of SAMA with multiple local searches that employs different metamodel (MSAMAs), which are listed in Table 4. For clarity, the differences between ESAMA and MSAMA are illustrated in Figure 6.

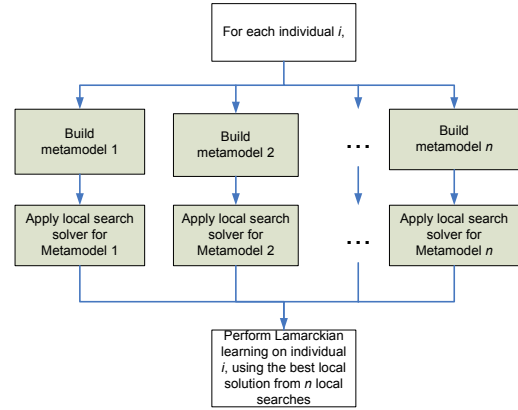
Scheme	No. of Local Searches per Individual	No. of Exact Evaluations per Individual (for trust region iteration, $k=3$)
ESAMA-GPPR	1(GPPR)	3
MSAMA-GP+PR	2(GP, PR)	6
MSAMA-GPPR+GP	2(GPPR, GP)	6
MSAMA-GPPR+PR	2(GPPR, GP)	6
MSAMA-GPPR+GP+PR	3(GPPR, GP, PR)	9

Table 4: The different combinations of using multiple metamodels in MSAMA search.

The ESAMA considered here is a variant of SAMA which utilizes a surrogate ensemble of GP and PR models and denoted as ESAMA-GPPR. The two models are chosen as they have shown to be more robust than others based on our results in section 4.1. In GPPR ensemble, every approximated value is obtained from an aggregated model $M(x)$, constructed from the average of GP and PR models. MSAMA-GP+PR represents a two-model MSAMA, hence two local searches are performed for each individual, i.e. one local search per model. Note that surrogate ensemble may also be considered as one of the independent surrogate models used in MSAMA. Hence, the mixture of ESAMA and MSAMA are represented as MSAMA-GPPR+GP and MSAMA-GPPR+PR schemes, where one local search is performed using an ensemble while another uses either GP or PR model alone. The last scheme considered is MSAMA-GPPR+GP+PR, which consumes a higher computational effort per individual evaluation, among all five considered, since three local searches are performed for every individual. Note that since the trust region iteration $k=3$, each local search in



(a) ESAMA



(b) MSAMA

Figure 6: Comparison between ESAMA and MSAMA.

any scheme will incur three exact function evaluations per individual.

The convergence trends for optimizing the four test functions are presented in Figures 7-10. The figures show that in most cases, SAMA schemes using multiple metamodels, i.e. MSAMA and/or ESAMA, generally perform better than those counterparts that employs only a single metamodel. For the Ackley, Griewank, and Step functions, MSAMA-GPPR+PR scheme results in best convergence search trends. SAMA seems to leverage well from synergies between GPPR ensemble and PR models in these cases. On the other hand, as three extra exact evaluations are required in MSAMA-GPPR+GP+PR scheme, in most cases, this scheme did not materialize when compared to others for the same computational budget.

From our investigations, it should be noted that while some mild assumption on the accuracy of the metamodeling technique used is necessary for SAMA, the capability to smooth the function is also useful for achieving improved or faster convergence rate. While some metamodeling tech-

niques perform best on particular benchmark problems, ensemble of metamodels and multi-surrogates yield robust and improved solution quality on the benchmark problems in general, for the same computational budget. Hence, using ensembles of metamodels and/or multiple metamodels are indeed more effective in SAMAs.

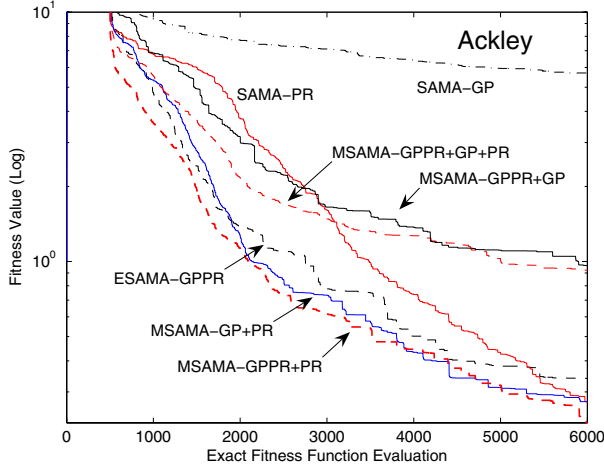


Figure 7: Convergence trends of using different combinations of GP and PR models in SAMA for Ackley function.

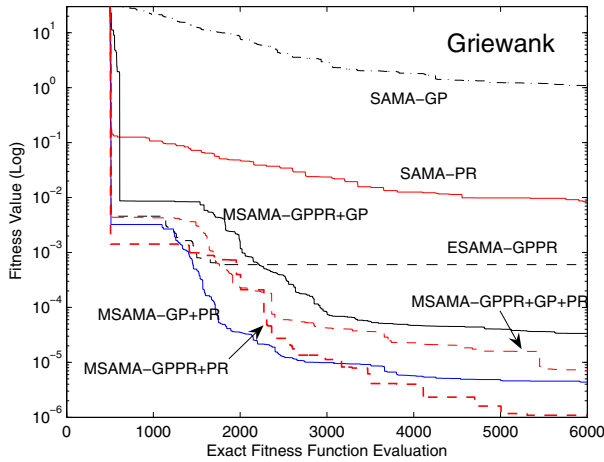


Figure 8: Convergence trends of using different combinations of GP and PR models in SAMA for Griewank function.

5. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the effects of utilizing different metamodels in Surrogate-Assisted Memetic Algorithm with Trust Region Framework (SAMA-TRF), a variant of SAMA which incorporates a trust region framework for interleaving use of exact objective function with computationally cheap local approximation during local searches. Four different metamodels, namely Gaussian Process (GP),

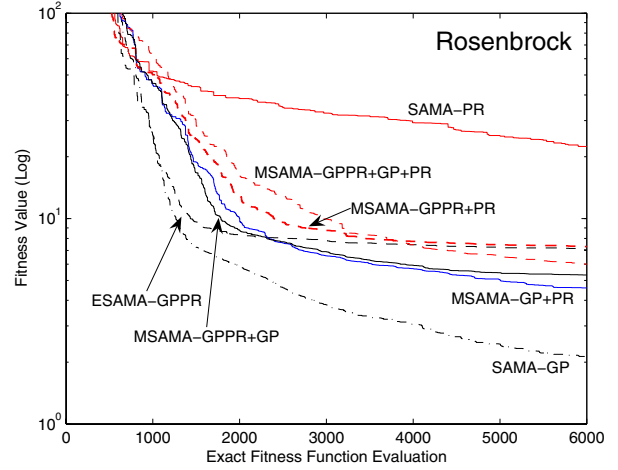


Figure 9: Convergence trends of using different combinations of GP and PR models in SAMA for Rosenbrock function.

Radial Basis Function (RBF), Polynomial Regression (PR), and Extreme Learning Machine (ELM) neural network are used in the study. Empirical results show that while all metamodels used result in search improvement compared to standard GA, some metamodels, such as PR and GP do show great robustness when applied to all test functions. A study of different schemes which utilize multiple metamodels in SAMA-TRF has also been presented and shows promising results for further study. In our future work, more study will be performed on constructing a robust model management framework which can take advantage from all the different metamodeling techniques through the use of multiple metamodels in SAMA in particular, and SAEA in general.

6. ACKNOWLEDGMENTS

D. Lim and Y.S. Ong would like to thank Honda Research Institute Europe, Germany for funding this research work and the members of Nanyang Technological University, Singapore for providing the computing resources and supports.

7. REFERENCES

- [1] M.K. Karakasis, A.P. Giotis, and K.C. Giannakoglou. Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization. *International Journal for Numerical Methods in Fluids*, 43, pp. 1149 - 1166, 2003.
- [2] Y. S. Ong and A.J. Keane. A domain knowledge based search advisor for design problem solving environments. *Engineering Applications of Artificial Intelligence*, 15(1), pp. 105-116, 2002.
- [3] M. Olhofer, T. Arima, T. Sonoda, and B. Sendhoff. Optimization of a stator blade used in a transonic compressor cascade with evolution strategies. *Adaptive Computing in Design and Manufacture (ACDM)*, Springer Verlag, pp. 45-54, 2000.
- [4] K. C. Tan, R. Sathikannan, W. W. Tan, and A. P. Loh. Evolutionary design and implementation of a hard disk drive servo control system. *Soft Computing*, 11(2), pp. 131-139, 2007.

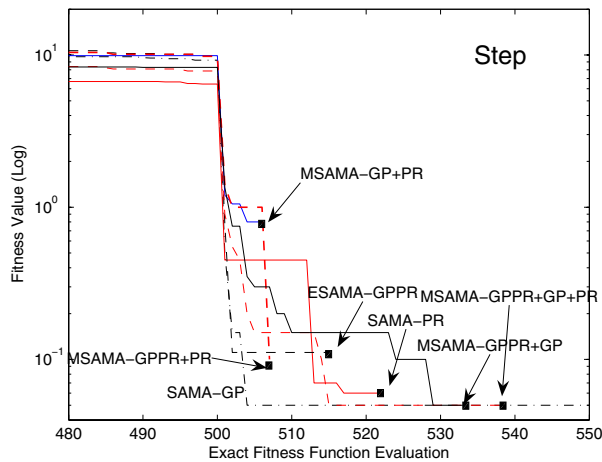


Figure 10: Convergence trends of using different combinations of GP and PR models in SAMA for Step function.

[5] K. C. Tan, E. F. Khor, J. Cai, C. M. Heng, and T. H. Lee. Automating the drug scheduling of cancer chemotherapy via evolutionary computation. *Artificial Intelligence in Medicine*, 25, pp. 169-185, 2002.

[6] A. Ratle. Kriging as a surrogate fitness landscape in evolutionary optimization. *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, 15(1), pp. 37-49, 2001.

[7] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1), pp. 3-12, 2005.

[8] M.A. El-Beltagy, P.B. Nair, A.J. Keane. Metamodeling techniques for evolutionary optimization of computationally expensive problems: promise and limitations. *Proc. of the GECCO 2003*, pp. 196-203, 1999.

[9] Y. Jin, M. Olhofer, B. Sendhoff. A framework for evolutionary optimization with approximate fitness function. *IEEE Transactions on Evolutionary Computation*, 6(5), pp. 481-494, October 2002.

[10] H. Ulmer, F. Streichert, and A. Zell. Evolution strategies assisted by gaussian processes with improved pre-selection criterion. *Proc. of IEEE CEC 2003*, pp. 692-699, 2003.

[11] Y.S. Ong, P.B. Nair, and A.J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA Journal*, 41(4), pp. 687-696, 2003.

[12] J. F. Rodriguez, J. E. Renaud, and L. T. Watson. Convergence of trust region augmented lagrangian methods using variable fidelity approximation data. *Structural Optimization*, 15(3-4), pp. 141-156, 1998.

[13] Y.S. Ong, P.B. Nair, K.Y. Lum. Evolutionary algorithm with hermite radial basis function interpolants for computationally expensive adjoint solvers. *Computational Optimization and Applications*, 2006. (In Press)

[14] Z. Zhou, Y.S. Ong, P.B. Nair, A.J. Keane, and K.Y. Lum. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. *IEEE Transactions on Systems, Man and Cybernetics (SMC), part C*, 37(1), pp. 66-76, Jan 2007.

[15] Y.S. Ong and A.J. Keane. Meta-lamarckian learning in memetic algorithm. *IEEE Transactions on Evolutionary Computation*, 8(2), pp. 99-110, 2004.

[16] Y.S. Ong, P.B. Nair, and A.J. Keane, and K.W. Wong. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation, Studies in Fuzziness and Soft Computing*, pp. 307-332. 2004.

[17] N. Alexandrov, J.E. Dennis, R.M. Lewis, and V. Torczon. A trust region framework for managing the use of approximation models in optimization. *Journal on Structural Optimization*, 15(1), pp. 16-23, 1998.

[18] C.T. Lawrence and A.L. Tits. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software*, 6, pp. 265-282, 1996.

[19] D. J. C. Mackay. Introduction to gaussian processes. *Neural Networks and Machine Learning*, 168, pp. 133-165, 1998.

[20] C. Bishop. Neural networks for pattern recognition. Oxford University Press, 1995.

[21] Fred H. Lesh. Multi-dimensional least-square polynomial curve fitting. *Communications of ACM*, 2(9), pp. 29-30, 1959.

[22] G.B. Huang, Q.Y. Zhu, C.K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. *Proc. of IJCNN 2004*, 2, pp. 985-990, 2004.

[23] S. D. Handoko, C. K. Kwok, Y. S. Ong, L. Z. Guang and B. Vladimir. Extreme learning machine for predicting HLA-peptide binding. *Third ISNN*, Chengdu, May 28 - June 1, LNCS 3973, pp. 716-721, 2006.

[24] K. H. Liang, X. Yao, and C. Newton. Evolutionary search of approximated n -dimensional landscapes. *International Journal of Knowledge-Based Intelligent Engineering Systems*, vol. 4, no. 3, pp. 172183, 2000.

[25] Y. S. Ong, Z. Zhou, and D. Lim. Curse and blessing of uncertainty in evolutionary algorithm using approximation. *Proc. of The 2006 IEEE CEC 2006*, pp. 2928-2935, Vancouver, July 2006.

[26] R. Jin, W. Chen and T. W. Simpson, 2001. Comparative studies of metamodeling techniques under multiple modeling criteria. *Journal of Structural Optimization*, Vol. 23, No. 1, pp. 1-13, 2001.

[27] Z. Z. Zhou, Y. S. Ong, M. H. Lim and B. S. Lee. Memetic Algorithm using Multi-Surrogates for Computationally Expensive Optimization Problems. *Soft Computing Journal*, ISSN 1432-7643 (Print) 1433-7479 (Online), December 2006.

[28] Y. Jin and B. Sendhoff. Reducing fitness evaluations using clustering techniques and neural networks ensembles. *GECCO 2004*, LNCS 3102, pp. 688-699, 2004.