Evolutionary Optimization with Hierarchical Surrogates

Xiaofen Lu, *Student Member, IEEE*
Supervisors: Ke Tang and Xin Yao

***Abstract*—Using surrogate models in evolutionary search provides an effective means for Evolutionary Algorithms (EAs) to handle complex engineering design problems such as computationally expensive problems under limited computational resources. As different modeling techniques may model differently on different problem landscapes, the choice of modeling technique can further affect the performance of the evolutionary search. However, it is very hard to make an appropriate choice of modeling technique without any prior knowledge of the optimization problem. To address this issue, recent surrogate-assisted evolutionary frameworks have relied on simultaneous use of multiple modeling techniques or selecting one according to some performance metric. In this work, we consider a novel scheme to adapt the surrogate modeling technique in the evolutionary search process, which differs from existing approaches in employing a hierarchical structure of surrogates. The experimental results showed the superiority of our proposed algorithm over two state-of-the-art surrogate-assisted EAs addressing this issue.**

***Index Terms*—evolutionary algorithms, computationally expensive problems, surrogate model, multiple modeling techniques**

I. Introduction

As a class of stochastic global optimization algorithms, Evolutionary Algorithms (EAs) have become one of the most popular optimization techniques and achieved great success on a variety of real-world applications, such as music composition [1], financial forecasting [2], aircraft design [3], job shop scheduling [4], and drug design [5].

However, new challenges still arise for EAs due to increasing computational needs in real-world applications. For instance, a continuing trend in the engineering field is the use of increasingly high-fidelity analysis codes such as Computational Structural Mechanics (CSM), Computational Electro-Magnetics (CEM), and Computational Fluid Dynamics (CFD) in the design and simulation process to evaluate the system performance of one design. This brings a class of Computationally Expensive Problems (CEPs) for which evaluating the quality of a candidate solution (i.e., one fitness evaluation) may take from minutes to hours of supercomputer time. For example, one function evaluation involving the solution of the Navier-Stokes equations can take many hours of computer time in aerodynamic wing design [6]. As EAs usually need a lot of fitness evaluations to achieve a satisfying solution, it becomes computationally prohibitive to use EAs to solve such problems.

In this context, researchers have developed many methods to make EAs suitable for CEPs. Among them, the use of surrogate models to replace the real fitness function within the evolutionary framework is becoming a common practice. Surrogate models are computationally efficient models, and can be used in lieu of the real fitness function to reduce computational cost [7]. For example, surrogate models can be interpolation or regression models that are built to approximate the real fitness function using some input output pairs evaluated by the fitness function. Through the use of surrogate models, the computational burden can be greatly reduced since the efforts required to build the surrogates and to use them are much lower than evaluating a candidate solution with the exact fitness function. Among the modeling techniques, multivariate polynomial regression method (PR), radial basis function (RBF) network method, Kriging method and support vector machines (SVMs) method are the most prominent and commonly used [8].

However, modeling techniques may model differently on different problem landscape [9]. Depending on the complexity of the optimization problem, the choice of modeling technique can further affect the performance of surrogate-assisted optimization. Given prior knowledge of the fitness landscape is unavailable beforehand, it is almost impossible to know which modeling technique is the most relevant for the fitness landscape or can generate reliable fitness predictions, which has become one of the greatest barriers to further progress in surrogate-assisted optimization. In the literature, various studies have been carried out along this direction.

Some studies focused on developing performance or assessment metrics to measure the performance of surrogate model. Particularly, the focus has been placed on building multiple surrogate models and selecting the best one that has minimum training error on the training points. Maximum/mean absolute error, root mean square error (RMSE) and correlation measure denote some of the performance metrics that are commonly used [6]. Typical model selection schemes that stem from the field of statistical and machine learning, including the split sample approach, cross-validation and bootstrapping were also employed to select surrogate models that have low estimation of apparent and true errors [10], [11]. In [12], multiple cross-validation schemes are used for the selection of low-error

This research is funded by the IEEE Computational Intelligence Society Graduate Student Research Grant 2013.

X. Lu, K. Tang, and X. Yao are with the Nature Inspired Computation and Applications Laboratory, USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications, School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: xiaofen@mail.ustc.edu.cn)

X. Yao is also with the Center of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom.

surrogates that replace the original costly high-fidelity analysis solver to avoid convergence at false optima of poor accuracy models.

In the field of multidisciplinary optimization, this issue is commonly handled by performing multiple optimization runs, each on a different surrogate model [13], [14] or ensemble model [15], [16]. Use of multiple surrogates was also considered in Efficient Global Optimization (EGO). In EGO, one optimization run is performed on the surrogate model to obtain a solution point per optimization cycle. The surrogate model will be updated with the obtained solution point and the updated model will be used in the next optimization cycle. In a recent work [17], the authors proposed generating several solution points by performing multiple optimization runs on multiple surrogate models per optimization cycle. Under the condition of parallel computing, this strategy exhibited superior performance than using single surrogate model. This is because that adding multiple solution points can make the surrogate model more accurate in the next optimization cycle but the total time of each optimization cycle does not increase when multiple optimization runs are performed in parallel.

Simultaneous use of multiple modeling techniques also exists in the evolutionary framework. In [9], a generalized evolutionary framework called GSM was proposed by unifying diverse surrogate models synergistically in the memetic evolutionary search. Specifically, for each individual in the current population, two surrogate models are constructed on the selected nearest evaluated points using different modeling techniques. One surrogate model is an ensemble of three models, which include the interpolating Kriging/Gaussian process (GP), interpolating linear RBF and second-order PR. The other is second-order PR. Then, one local search run is performed on each surrogate model to obtain an improved solution on the surrogate model. The two generated solutions will be evaluated with the real fitness function and the best one will be selected to replace the current individual if it is better than the current individual.

In the same memetic evolutionary framework, Le et al. proposed the Evolvability Learning of Surrogates (EvoLS) [18]. The concept of "evolvability" indicates the productivity of a modeling technique that brings about fitness improvement in the local search and was used as the basis for adapting modeling technique in the evolutionary search. The evolvability of one model technique is calculated based on the quality of the solutions found by the model technique in previous local search performed for individuals in previous generations. Surrogate models used in EvoLS include interpolating Kriging/Gaussian process (GP), interpolating linear RBF and second-order PR. A numerical study of EvoLS on several test functions showed the superiority of EvoLS over GSM.

In this paper, we propose a novel approach to adapt surrogate modeling techniques in the memetic evolutionary search process. Inspired from the hierarchical mixture of expert model [19], we consider adapting the surrogate modeling through a two-level hierarchical structure. In our approach, for each individual in the current generation, several surrogate models are built as the low-level surrogate models. Then, local search is performed on each low-level surrogate model to find an improved solution over the surrogate model. After this, a high-level surrogate model is built to identify the potentially best one among the newly found solutions, and meanwhile the most promising low-level surrogate model is identified. In the end, only this identified solution is evaluated with the exact fitness function.

Compared to the method employed in GSM, our method can be cost effective as GSM needs multiple fitness evaluations to select the best surrogate model while our method only need one fitness evaluation. In comparison with EvoLS, EvoLS actually selects modeling technique through predicting its current performance based on its past performance(in the calculation of evolvability, EvoLS assumes the same modeling technique now generates the same solution in the local search for one individual as it does for the same individual before). However, our method can rely on prediction the fitness or rank of solutions found on each surrogate model, which seems more reliable than the performance prediction of surrogate modeling because the performance of surrogate model changes as more sampling points are generated. In this paper, the performance of GSM, EvoLS and our our algorithm, evolutionary optimization with hierarchical surrogates (EHS in brief), will be statistically compared considering the quality of the obtained solutions on several benchmark functions.

The rest of this paper is organized as follows. In Section II, details of the proposed algorithm is given. In Section III, experimental results and analysis are presented to evaluate the efficacy of EHS. Finally, Section IV concludes this paper.

II. The proposed approach

In this section, the proposed approach to adapt surrogate modeling techniques in the memetic evolutionary search process was proposed.

Without loss of generality, we assume in this paper the optimization problem has the following formulation.

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is a vector of n decision variables in a continuous decision space $\Omega = \prod_{i=1}^{n}[l_i, u_i]$, and $f : \Omega \subseteq \Re^n \to \Re$ is called the objective function.

Assuming the number of surrogate modeling techniques is k, the outline of the generated algorithm (EHS) can be shown as in Algorithm 1.

The EHS begins with the initialization of a population of candidate solutions. During the database building phase, the population is evolved with selection, crossover and mutation operators using exact fitness evaluations for a certain number of generations, and all exact evaluations are archived into a database DB. This is to accumulate training samples to build surrogate models.

Subsequently, EHS proceeds in the phase during which surrogate models are involved. For each individual in the current generation, m nearest evaluated points to it according to the Euclidian distance are selected as the training set. Then, k low-level surrogate models are trained on this set to approximate the fitness function using different approximation methodologies. For each low-level surrogate model, local search is performed to find one improved solution over the surrogate model. After this, a high-level surrogate model is built to identify the potentially best one among the newly found solutions. Then, this identified solution is evaluated with the exact fitness function and this solution will replace the current individual if it is better than the current individual. This process iterates until the computational budget is exhausted, i.e., all the fitness evaluations are used up.

Algorithm 1 EHS

Initialize a population $\boldsymbol{P}_G = \{\boldsymbol{x}_{i,G} | i = 1, 2, ..., popsize\}$
Evaluate $\boldsymbol{P}_G$ using the exact objective function
Archive all exact evaluations into a database DB
while computational budget is not exhausted **do**
  if database building phase does not end **then**
    Evolve $\boldsymbol{P}_G$ with evolutionary operators (selection, crossover and mutation) using exact evaluations, archive all exact evaluations into DB
  else
    Apply evolutionary operators to create a new population $\boldsymbol{P}_G$
    for each $\boldsymbol{x}_{i,G}$ in $\boldsymbol{P}_G$ **do**
      Find m nearest points to $\boldsymbol{x}_{i,G}$ in DB as training points for surrogate models.
      Training k low-level surrogate models on these points using different modeling methodologies
      Apply local search in each surrogate model to arrive at k new solutions.
      Build a high-level surrogate model S based on DB and the k new solutions
      Select the best one among the k solutions according to S (denoted as $\boldsymbol{x}_{i,G}^{opt}$)
      if f($\boldsymbol{x}_{i,G}^{opt}$)<f($\boldsymbol{x}_{i,G}$) **then**
        $\boldsymbol{x}_{i,G}$=$\boldsymbol{x}_{i,G}^{opt}$
      end if
      Archive all exact evaluations into DB
    end for
  end if
  Set $G = G + 1$
end while

A. Low-level surrogate models

For each current individual, several low-level surrogate models are trained using the nearest evaluated points in the database to approximate the fitness function. These surrogate models are made different from each other through the use of different modeling techniques such as PR, RBF, Kriging and so on. They can also be created using the same modeling technique with different modeling parameter settings.

B. Local Search

For each individual $\boldsymbol{x}_{i,G}$ in the population, the local search method is to minimize each problem on the landscapes approximated by the low-level surrogate models. The problem on the landscape approximated by the j-th low-level surrogate model has the form:

$$\begin{aligned} &\min \hat{f}_j(\boldsymbol{x}_{i,G} + \boldsymbol{s}) \\ &\text{subject to: } \| \boldsymbol{s} \| \leq \Delta \end{aligned} \tag{2}$$

where $\Delta = [\min_{i=1,2,...,m} \{\boldsymbol{y}_{(i)}^{(k)}\}, \max_{i=1,2,...,m} \{\boldsymbol{y}_i^{(k)}\}]_{k=1,2,...,n}$, $\forall i = 1, 2, ..., m$, and $\boldsymbol{y}_i^{(k)}$ denotes the k-th dimension of the i-th selected training sample.

In this paper, an efficient local search strategy, Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B) method [20], is employed to minimize the problem on each approximated landscape.

C. High-level surrogate model

The high-level surrogate model is built to select best one among the solutions found by the local search over each surrogate model. For high-level surrogate models, we consider ranking models in this paper as they seems more appropriate to select the best individuals [21], [22], and use an efficient algorithm RankBoost [23] to build ranking models.

Furthermore, to build a ranking model, we select m' nearest evaluated points in the database for each solution found over low-level surrogate models and combine them together, 80% of which are chosen uniformly as the training set and the remaining 20% form the set for validating the prediction quality. If the prediction accuracy of S is larger than 0.5 (the accuracy of a random approach), the solution appears the best according to S will be selected to evaluate with the exact objective function, otherwise a solution is randomly selected for the exact evaluation.

III. Experimental Validation

To validate the efficiency of the proposed EHS, experimental studies have been conducted to compare EHS with GSM and EvoLS. In this section, numerical results on the proposed EHS are obtained using the same three approximation methodologies (i.e., $k = 3$) as used in [9] and [18], i.e., 1) interpolating linear spline RBF, 2) second order PR and 3) interpolating Kriging/GP. Details of PR, RBF and Kriging/GP can be found in [18].

A. Experimental Setup

Empirical study on the EHS was performed using the same 10 test functions used in GSM (6 of which were used in EvoLS). The 10 benchmark functions (F1-F10) were reported in [24], [25] and summarized in Table I. More detailed description of these functions can be found in [24], [25]. In this paper, the number of decision variables, n, was set to 30 for all benchmark functions.

Considering that only limited computational resources are allowable to solve CEPs, for each test function, the maximum number of fitness evaluations is set to 3000 and the maximum number of fitness evaluations in the building phase is set to 600. For each of GSM, EvoLS and EHS, the number of nearest points selected for each current individual is set

TABLE I
THE BENCHMARK FUNCTIONS

Benchmark Function	Description	Global Optimum
F1	Ackley	0.0
F2	Griewank	0.0
F3	Rosenbrock	0.0
F4	Shifted Rotated Rastrigin (F10 in [25])	-330.0
F5	Shifted Rotated Weierstrass (F11 in [25])	90.0
F6	Shifted Expanded Griewank plus Rosenbrock (F13 in [25])	-130.0
F7	Hybrid Composition Function (F15 in [25])	120.0
F8	Rotated Hybrid Composition Function (F16 in [25])	120.0
F9	Rotated Hybrid Composition Function with Narrow Basin Global Optimum (F19 in [25])	10.0
F10	Noncontinuous Rotated Hybrid Composition Function (F23 in [25]	360.0

to 200 (i.e., $m = 200$ for EHS). The value of m' used in building the high-level surrogate model in EHS is set to 100. The other parameters for GSM and EvoLS were set the same as in their original papers. The other parameter configurations of EHS are defined as in Table II. Note that the basis evolutionary algorithm of EHS is Genetic Algorithm (GA) and the parameter setting of it is the same as in GSM.

In our experiment, 25 independent runs were conducted for each algorithm on each test function in the MATLAB environment. The minimum objective function values found by each algorithm at 1000 FEs, 2000 FEs and 3000 FEs on each test function over the 25 runs were recorded to measure its performance. Moreover, Wilcoxon rank-sum test at a 0.05 significance level was used in this paper to compare EHS with each of GSM and EvoLS.

TABLE II
ALGORITHM CONFIGURATION

Parameter Settings	
Population size	100
Selection scheme	Elitism and ranking selection
Variation operators	Uniform crossover and mutation
Crossover probability	0.9
Mutation probability	0.1
Local search method	L-BFGS-B
Stopping criteria	3000 evaluations
Database building phase	600 evaluations
Number of independent runs	25

B. Experimental Results

Tables III and VIII summarize the average and standard deviation of the function error values obtained by the 3 algorithms on all the test functions at 1000 FEs, 2000 FEs and 3000 FEs. The results of Wilcoxon rank-sum test are presented in the last three rows of each table.

As can be seen from the last three rows of Tables III and V, EHS obtained higher solution quality than GSM on all the test functions at each of 1000 FEs, 2000 FEs and 3000FEs.

From the last three rows of Tables VI and VIII, it can be seen that EHS outperformed EvoLS on all the test functions

TABLE III
EXPERIMENTAL RESULTS OF GSM AND EHS OVER 25 RUNS AT 1000 FES ON 10 TEST FUNCTIONS OF 30 VARIABLES, $+$, $-$, AND $\approx$ DENOTE THAT THE RESULT OF GSM IS BETTER THAN, WORSE THAN, AND COMPARABLE TO THAT OF EHS, RESPECTIVELY

Func	GSM MeanError±StdDev	EHS MeanError±StdDev
F1	1.36e+001±1.29e+000 −	1.06e+001±6.01e-001
F2	5.41e-001±8.02e-001 −	3.55e-007±3.68e-007
F3	1.71e+002±4.43e+001 −	6.18e+001±1.21e+001
F4	-4.02e+001±2.04e+001 −	-9.42e+001±2.89e+001
F5	1.36e+002±1.19e+000 −	1.34e+002±1.73e+000
F6	-5.67e+001±1.88e+001 −	-1.01e+002±2.45e+000
F7	8.80e+002±6.96e+001 −	8.42e+002±4.60e+001
F8	5.50e+002±7.71e+001 −	4.86e+002±5.83e+001
F9	1.10e+003±2.20e+001 −	1.07e+003±1.53e+001
F10	1.56e+003±3.06e+001 −	1.49e+003±3.78e+001
−	10	
+	0	
≈	0	

TABLE IV
EXPERIMENTAL RESULTS OF GSM AND EHS OVER 25 RUNS AT 2000 FES ON 10 TEST FUNCTIONS OF 30 VARIABLES, $+$, $-$, AND $\approx$ DENOTE THAT THE RESULT OF GSM IS BETTER THAN, WORSE THAN, AND COMPARABLE TO THAT OF EHS, RESPECTIVELY

Func	GSM MeanError±StdDev	EHS MeanError±StdDev
F1	9.16e+000±1.21e+000 −	6.77e+000±3.66e-001
F2	3.97e-001±5.29e-001 −	2.51e-007±2.48e-007
F3	9.13e+001±1.55e+001 −	2.75e+001±7.63e-001
F4	-7.52e+001±2.28e+001 −	-2.68e+002±1.43e+001
F5	1.35e+002±1.09e+000 −	1.28e+002±5.17e+000
F6	-7.92e+001±1.04e+001 −	-1.09e+002±1.22e+000
F7	7.76e+002±7.21e+001 −	6.30e+002±4.68e+001
F8	4.58e+002±6.83e+001 −	2.95e+002±8.59e+001
F9	1.04e+003±1.81e+001 −	1.02e+003±2.00e+001
F10	1.45e+003±7.81e+001 −	1.21e+003±1.40e+002
−	10	
+	0	
≈	0	

TABLE V
EXPERIMENTAL RESULTS OF GSM AND EHS OVER 25 RUNS WITH 3000 FES ON 10 TEST FUNCTIONS OF 30 VARIABLES, $+$, $-$, AND $\approx$ DENOTE THAT THE RESULT OF GSM IS BETTER THAN, WORSE THAN, AND COMPARABLE TO THAT OF EHS, RESPECTIVELY

Func	GSM MeanError±StdDev	EHS MeanError±StdDev
F1	5.50e+000±6.24e-001 −	5.11e+000±4.79e-001
F2	3.62e-001±4.76e-001 −	2.04e-007±1.44e-007
F3	4.93e+001±8.53e+000 −	2.66e+001±8.18e-001
F4	-9.69e+001±2.16e+001 −	-2.75e+002±1.26e+001
F5	1.35e+002±9.18e-001 −	1.21e+002±7.46e+000
F6	-8.93e+001±6.32e+000 −	-1.11e+002±1.26e+000
F7	7.24e+002±6.46e+001 −	5.81e+002±3.97e+001
F8	4.41e+002±6.73e+001 −	2.85e+002±8.91e+001
F9	1.03e+003±1.44e+001 −	9.84e+002±3.14e+001
F10	1.38e+003±1.17e+002 −	1.07e+003±1.18e+002
−	10	
+	0	
≈	0	

TABLE VI
EXPERIMENTAL RESULTS OF EVOLS AND EHS OVER 25 RUNS WITH 1000 FES ON 10 TEST FUNCTIONS OF 30 VARIABLES, $+$, $-$, AND $\approx$ DENOTE THAT THE RESULT OF EVOLS IS BETTER THAN, WORSE THAN, AND COMPARABLE TO THAT OF EHS, RESPECTIVELY

Func	GSM MeanError±StdDev	EHS MeanError±StdDev
F1	1.38e+001±1.27e+000 $-$	1.06e+001±6.01e-001
F2	7.18e-001±1.92e+000 $-$	3.55e-007±3.68e-007
F3	2.47e+002±7.43e+001 $-$	6.18e+001±1.21e+001
F4	-3.01e+001±3.04e+001 $-$	-9.42e+001±2.89e+001
F5	1.36e+002±2.18e+000 $-$	1.34e+002±1.73e+000
F6	-5.48e+001±2.80e+001 $-$	-1.01e+002±2.45e+000
F7	9.96e+002±5.83e+001 $-$	8.42e+002±4.60e+001
F8	6.71e+002±1.23e+002 $-$	4.86e+002±5.83e+001
F9	1.18e+003±2.91e+001 $-$	1.07e+003±1.53e+001
F10	1.63e+003±2.43e+001 $-$	1.49e+003±3.78e+001
$-$	10	
$+$	0	
$\approx$	0	

TABLE VII
EXPERIMENTAL RESULTS OF EVOLS AND EHS OVER 25 RUNS WITH 2000 FES ON 10 TEST FUNCTIONS OF 30 VARIABLES, $+$, $-$, AND $\approx$ DENOTE THAT THE RESULT OF EVOLS IS BETTER THAN, WORSE THAN, AND COMPARABLE TO THAT OF EHS, RESPECTIVELY

Func	GSM MeanError±StdDev	EHS MeanError±StdDev
F1	5.92e+000±7.76e-001 $+$	6.77e+000±3.66e-001
F2	2.40e-001±4.32e-001 $-$	2.51e-007±2.48e-007
F3	1.20e+002±2.04e+001 $-$	2.75e+001±7.63e-001
F4	-7.06e+001±2.31e+001 $-$	-2.68e+002±1.43e+001
F5	1.36e+002±1.45e+000 $-$	1.28e+002±5.17e+000
F6	-8.52e+001±1.18e+001 $-$	-1.09e+002±1.22e+000
F7	8.38e+002±6.33e+001 $-$	6.30e+002±4.68e+001
F8	4.78e+002±7.70e+001 $-$	2.95e+002±8.59e+001
F9	1.06e+003±2.40e+001 $-$	1.02e+003±2.00e+001
F10	1.55e+003±4.06e+001 $-$	1.21e+003±1.40e+002
$-$	9	
$+$	1	
$\approx$	0	

TABLE VIII
EXPERIMENTAL RESULTS OF EVOLS AND EHS OVER 25 RUNS WITH 3000 FES ON 10 TEST FUNCTIONS OF 30 VARIABLES, $+$, $-$, AND $\approx$ DENOTE THAT THE RESULT OF EVOLS IS BETTER THAN, WORSE THAN, AND COMPARABLE TO THAT OF EHS, RESPECTIVELY

Func	GSM MeanError±StdDev	EHS MeanError±StdDev
F1	2.67e+000±4.02e-001 $+$	5.11e+000±4.79e-001
F2	2.16e-001±3.76e-001 $-$	2.04e-007±1.44e-007
F3	5.30e+001±1.15e+001 $-$	2.66e+001±8.18e-001
F4	-7.59e+001±2.70e+001 $-$	-2.75e+002±1.26e+001
F5	1.37e+002±1.69e+000 $-$	1.21e+002±7.46e+000
F6	-8.80e+001±1.45e+001 $-$	-1.11e+002±1.26e+000
F7	7.73e+002±5.46e+001 $-$	5.81e+002±3.97e+001
F8	4.45e+002±6.86e+001 $-$	2.85e+002±8.91e+001
F9	1.04e+003±1.91e+001 $-$	9.84e+002±3.14e+001
F10	1.52e+003±7.42e+001 $-$	1.07e+003±1.18e+002
$-$	9	
$+$	1	
$\approx$	0	

at 1000 FEs, 9 test functions at 2000 FEs and 9 test functions at 3000 FEs, while EvoLS only performed better than EHS on one test function at each of 2000 FEs and 3000 FEs.

Overall, EHS can give solutions of higher quality than GSM and EvoLS with a specified number of fitness evaluations. Thus, EHS is more appropriate for CEPs, for which only limited computational resources available.

Moreover, when checking the true rank of the surrogate model selected by EHS each time, i.e., the rank of the solution generated through performing local search on this selected surrogate model among all the solutions generated on all the three surrogate models, and the true rank of the the surrogate model selected by EvoLS in its framework, it was found that EHS overall can achieve higher average rank than EvoLS does. This can substantiate our previous claim that the selection strategy of EHS is more reliable than that of EvoLS.

IV. CONCLUSION

Surrogate models are usually incorporated in EAs to better solve CEPs. In surrogate-assisted evolutionary search, the choice of surrogate modeling technique can affect the performance of the evolutionary search. To address this issue, we proposed a novel scheme in this work to adapt the surrogate modeling technique in the memetic evolutionary search process. The scheme differs from existing approaches that try to address this issue in employing a hierarchical structure of surrogates. The generated algorithm is called EHS. To validate the efficiency of EHS, we conducted experiments on 10 commonly used benchmark functions with a maximum number of fitness evaluations of 3000 and two state-of-the-art surrogate-assisted EAs (i.e., GSM and EvoLS) were used for making comparisons. The experimental results showed that EHS can overall achieved solution of higher quality than GSM and EvoLS do on all the test functions at each of 1000 FEs, 2000 FEs, and 3000 FEs. Thus, EHS is more appropriate for solving CEPs.

ACKNOWLEDGMENT

The present research work was partially supported by IEEE Computational Intelligence Society Graduate Student Research Grant 2013.

REFERENCES

[1] C. Johnson and J. Cardalda, “Genetic algorithms in visual art and music,” *Leonardo*, vol. 35, no. 2, pp. 175–184, 2002.

[2] C. Aranha and H. Iba, “The memetic tree-based genetic algorithm and its application to portfolio optimization,” *Memetic Computing*, vol. 1, no. 2, pp. 139–151, 2009.

[3] D. Simon, “Biogeography-based optimization,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, 2008.

[4] S. Hasan, R. Sarker, D. Essam, and D. Cornforth, “Memetic algorithms for solving job-shop scheduling problems,” *Memetic Computing*, vol. 1, no. 1, pp. 69–83, 2009.

[5] E. Lameijer, T. Bäck, J. Kok, and A. Ijzerman, “Evolutionary algorithms in drug design,” *Natural Computing*, vol. 4, no. 3, pp. 177–243, 2005.

[6] Y. Ong, P. Nair, and A. Keane, “Evolutionary optimization of computationally expensive problems via surrogate modeling,” *AIAA journal*, vol. 41, no. 4, pp. 687–696, 2003.

[7] Y. Ong, P. Nair, A. Keane, and K. Wong, “Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems,” in *Knowledge Incorporation in Evolutionary Computation*. Springer, 2005, vol. 167, pp. 307–331.

[8] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft computing*, vol. 9, no. 1, pp. 3–12, 2005.

[9] D. Lim, Y. Jin, Y. Ong, and B. Sendhoff, “Generalizing surrogate-assisted evolutionary computation,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, 2010.

[10] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Tucker, “Surrogate-based analysis and optimization,” *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[11] Y. Tenne and S. Armfield, “Metamodel accuracy assessment in evolutionary optimization,” in *Proc. of the 2008 IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 1505–1512.

[12] ——, “A versatile surrogate-assisted memetic algorithm for optimization of computationally expensive functions and its engineering applications,” *Success in Evolutionary Computation*, vol. 92, pp. 43–72, 2008.

[13] L. Zerpa, N. Queipo, S. Pintos, and J. Salager, “An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates,” *Journal of Petroleum Science and Engineering*, vol. 47, no. 3, pp. 197–208, 2005.

[14] A. Samad, K. Kim, T. Goel, R. Haftka, and W. Shyy, “Multiple surrogate modeling for axial compressor blade shape optimization,” *Journal of Propulsion and Power*, vol. 24, no. 2, pp. 301–310, 2008.

[15] T. Goel, R. Haftka, W. Shyy, and N. Queipo, “Ensemble of surrogates,” *Structural and Multidisciplinary Optimization*, vol. 33, no. 3, pp. 199–216, 2007.

[16] E. Acar and M. Rais-Rohani, “Ensemble of metamodels with optimized weight factors,” *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, pp. 279–294, 2009.

[17] F. Viana, “Multiple surrogates for prediction and optimization,” *PhD Dissertation, University of Florida*, 2011.

[18] M. Le, Y. Ong, S. Menzel, Y. Jin, and B. Sendhoff, “Evolution by adapting surrogates,” *Evolutionary Computation*, vol. 21, no. 2, pp. 313–340, 2013.

[19] M. Jordan and R. Jacobs, “Hierarchical mixtures of experts and the em algorithm,” *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994.

[20] C. Zhu, L. P. Byrd, R.H., and J. Nocedal, “Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[21] T. Runarsson, “Ordinal regression in evolutionary computation,” *Proc. of the Parallel Problem Solving from Nature-PPSN IX*, pp. 1048–1057, 2006.

[22] I. Loshchilov, M. Schoenauer, and M. Sebag, “Comparison-based optimizers need comparison-based surrogates,” in *Proc. of the Parallel Problem Solving from Nature–PPSN XI*. Springer, 2011, pp. 364–373.

[23] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” *The Journal of machine learning research*, vol. 4, pp. 933–969, 2003.

[24] J. Digalakis and K. Margaritis, “On benchmarking functions for genetic algorithms,” *International journal of computer mathematics*, vol. 77, no. 4, pp. 481–506, 2001.

[25] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” *KanGAL Report*, vol. 2005005, 2005.