

# Fitness Landscape Approximation by Adaptive Support Vector Regression with Opposition-Based Learning

Pei, Yan  
Graduate School of Design, Kyushu University

Takagi, Hideyuki  
Faculty of Design, Kyushu University

<https://hdl.handle.net/2324/1905848>

---

出版情報 : 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013 : Proceedings, pp.1329-1334, 2013-10-16. IEEE

バージョン :

権利関係 :



# Fitness Landscape Approximation by Adaptive Support Vector Regression with Opposition-Based Learning

Yan PEI

Graduate School of Design  
Kyushu University  
Fukuoka, 815-8540 Japan  
peiyan@kyudai.jp

Hideyuki TAKAGI

Faculty of Design  
Kyushu University  
Fukuoka, 815-8540 Japan  
<http://www.design.kyushu-u.ac.jp/~takagi/>

**Abstract**—We propose a method for approximating a fitness landscape using adaptive support vector regression (SVR) with opposition based learning (OBL) to enhance the evolutionary search. This method tries to resolve the complexity of the fitness landscape in the original search space by designing a suitable kernel function with an adaptive parameter tuned by OBL; This kernel projects the original search space into a higher dimensional search space with a different topological structure. The elite is obtained from the approximated fitness landscape, using the adaptive SVR to accelerate the evolutionary computation (EC) search, and the individual with the worst fitness is replaced. The merits of the proposed method are evaluated by comparing it with the fitness landscape approximated in the original, in a lower and in a higher dimensional search space.

**Keywords**—support vector regression, evolutionary computation, fitness landscape, acceleration, adaptive parameter tuning, opposition-based learning

## I. INTRODUCTION

Improving evolutionary computation (EC) search capability has recently become a promising research topic in the computational intelligence community [3]. In general, the directions research is taking can be categorized according to three aspects. First is conducting different search strategies in accordance with knowledge obtained from the search process, such as fitness landscape approximation [4], [6], [7], [10], [12]. Second is to develop new strategies [8] or to search using a form of population-based hybrid global evolutionary algorithm coupled with a learning procedure capable of local refinements. Third is to develop a new EC algorithm inspired by a phenomenon or mechanism from physics, mathematics or biology [9]. All these approaches need to obtain the related knowledge from the problem, i.e. there must be a machine learning process embedded in the related algorithms.

Several methods for accelerating EC by regression or interpolation have been proposed. Reference [12] proposed to use a single peak function to approximate the fitness landscape to obtain an accelerated EC, however, its disadvantage is the high computational cost for  $n$ -dimensional regressions. Reference [2] extends this study by proposing to obtain the fitness landscape in each lower dimensional search space, and synthesize the obtained elite from the lower dimensional spaces

for search's next generation. From the comparative evaluation results [4], [10], we can see that computational cost, which also depends on problem complexity, can be saved.

Support vector machines (SVM) are supervised learning techniques based on the structural risk minimization principle [16]. It can be used in regression analysis to solve problems that are difficult to solve in their original domain or space, such as approximating a fitness landscape. The basic proposal of SVM is to project the original data into a higher dimensional space, i.e. a feature space, by some kernel function, and to conduct the regression in this feature space in the form of a linear combination of the training samples. These approaches belong to a class of algorithms which are called *Kernel Methods*.

Reference [17] proposed to use a SVM as a regression tool to approximate the fitness landscape and obtain an accelerated differential evolution (DE). This study considered SVR as a robust regression technique inherited from regularization techniques. However, for different landscapes or benchmark problems, it conducts the fitness landscape approximation using the same kernel function, i.e. the parameters of the kernel function are fixed. Because the topology and structure of the higher dimensional space are fixed by the kernel function with a fixed parameter, in some cases it may result in a worse regression. We need to adaptively approximate the fitness landscape with diverse kernel functions according to different tasks.

To address this limitation, we propose using an SVR technique with an adaptive parameter tuned by opposition-based learning to approximate the fitness landscape; this should resolve the landscape complexity and obtain knowledge about the corresponding global optimum. Kernel function design and utilization are the crucial issues in our proposed approach. In this study, we use a polynomial kernel in the regression process to evaluate the performance of benchmark functions. We also propose an algorithm framework which uses an OBL adaptive SVR to approximate the fitness landscape and thus enhance EC search capability.

Following this introduction, section II gives a brief review of support vector regression, kernel functions and the opposition-based learning in our study. An algorithm framework and search mechanism, i.e. surrogate-assisted evolutionary computation by adaptive SVR is described, and the

related surrogate model design and global region prediction are derived in section III. To determine the performance of our proposal, we evaluate it with a set of benchmark functions against our previous enhancement approaches in section IV. Some data analysis is then performed, problems with the design of our approach and open topics are discussed in section V. In section VI, the future works and conclusion of the whole paper are presented.

## II. A BRIEF INTRODUCTION TO SUPPORT VECTOR REGRESSION AND OPPOSITION-BASED LEARNING

### A. Support Vector Regression

Support vector regression (SVR) is a support vector machine (SVM)-based approach that is used in regression applications. If a problem has nonlinear characteristics, instead of trying to fit a nonlinear model directly, the SVR maps the problem first from the input space to a new higher dimensional space, i.e. a feature space, by conducting a nonlinear transformation using suitably chosen basis functions. Next, it uses a linear model in the feature space to solve the problem. There are two main aspects to using SVR; the first is to project the original space into a higher dimensional space such that linearity can be found, the other is to conduct the regression process using those linear characteristics.

$$X = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix} \quad (1)$$

$$\overline{X} = \begin{bmatrix} \overline{\phi(x_1)^T} \\ \vdots \\ \overline{\phi(x_N)^T} \end{bmatrix} = \begin{bmatrix} 1, & \phi(x_1)^T \\ & \vdots \\ 1, & \phi(x_N)^T \end{bmatrix} = [1_{N \times 1}, X] \quad (2)$$

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \quad (3)$$

Given some number of training sample data in the original space,  $x_1, x_2, \dots, x_N$ , there is a feature map function  $\phi(x_i)$ ,  $i = 1, 2, \dots, N$ , which maps the data into a higher dimensional space (Equation (1), (2) and (3)). From the normal equation of linear regression, we can obtain the support vector regression expression (Equation (4)).

$$f(x) = (\alpha_1 + \dots + \alpha_N) + \alpha_1 K(x_1, x) + \dots + \alpha_N K(x_N, x) \quad (4)$$

$$\alpha = \overline{X}(\overline{X}^T \overline{X})^{-2} \overline{X}^T y \quad (5)$$

### B. Kernel Function

The selection of kernel function is crucial for the success of all kernel algorithms, because the kernel constitutes the prior knowledge that is available about the task. Equation (6) is a polynomial kernel function. In our proposed evolution control approaches, we design a parametrically adaptive polynomial kernel function as shown in Equation (7).  $P_a$  is a parameter tuned by opposition-based learning which decides the topological structure of the higher dimensional search space.

$$K(x, z) = (< x, z > + 1)^r \quad (6)$$

$$K(x, z) = P_a * (< x, z > + 1)^r \quad (7)$$

### C. Opposition-Based Learning

Opposition-based learning (OBL) [14] is used in machine learning [15] and to accelerate optimization searches (OBL optimization). In OBL, the concept of an opposition number is used, as shown in Equation (8).

$$OP(x_i) = a + b - x_i \quad (8)$$

In our proposed SVR parameter learning process, if the fitness of an elite obtained by SVR with an opposite parameter is better than that obtained by the SVR with the original parameter, we use the opposite parameter in the next generation and update the parameter range until it is stable.

## III. EVOLUTIONARY COMPUTATION ACCELERATION BY ADAPTIVE SUPPORT VECTOR REGRESSION WITH OPPOSITION-BASED LEARNING

### A. Approximating the Fitness Landscape in Higher Dimensional Search Spaces

Our motivation is to resolve the fitness landscape complexity by projecting the individual into a higher dimensional search space by adaptive SVR; we can then find a simple landscape with which to search for the global optimum or predict its region. Equation (4) shows the regression model of the SVR. It is also the approximation model which we use to investigate the EC fitness landscape in this study.

Fig. 1 shows a visual illustration of our proposal. Sub-figure (a) is a fitness landscape in an original search space; although it is just a one dimensional search space, it is difficult to approximate or to fit in such one dimensional space. Sub-figure (b) is a fitness landscape that is projected into two dimensional search space; the dimensionality of the search space is increased, however, the landscape shape becomes smoother and more easily approximated.

In the kernel methods, which includes SVM, SVR, ..., there is no explicit feature map  $\phi(x_N)$ . The kernel function takes the role of projection, and determines the structure of higher dimensional search space and thus imparts the prior knowledge of the problem. The performance of our proposal depends upon two elements. One is the kernel function selection, the other is the parameter settings.

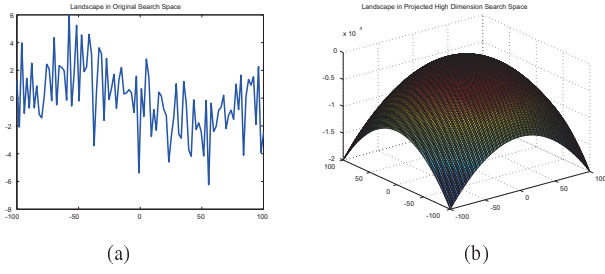


Fig. 1. Landscape in original search space (a) and after projection into the higher dimensional search space (b).

### B. Obtaining Elite from an Approximated Landscape by Designing an Adaptive Kernel Function

Equation (4) is the regression model in the higher dimensional search space. When a concrete kernel function is specified in the model, it implements a corresponding feature map and constructs a higher dimensional search space to approximate the fitness landscape. Analyzing the characteristics of the kernel function is thus crucial for finding the global optimum and predicting promising search regions.

The kernel function (Equation (6)) is a polynomial function, which makes the acquisition of its derivative information simple when the power is two, such that the inflection point will indicate the optimum in the search space. Because Equation (6) implements a fixed feature map, in our study we apply a polynomial function with a parameter that is adaptively tuned by OBL (Equation (7)). In the elite-finding process, we use the inflection point as the elite.

In our proposed SVR acceleration method, the elite is obtained from an approximated landscape curve, which is a crucial part of the process for enhancing the EC. After adaptive tuning of the kernel function's variable parameter, the global optimum information or the information concerning its predicted region can be obtained from the adaptive SVR modal. This presents the originality of our proposal, which tunes the parameter of the kernel functions in the SVR to approximate the fitness landscape. Another way of constructing a higher dimensional search space is to design a new kernel function for a specific concrete EC application. This should be a promising research direction for applications of our proposal, and it is also involved in our future work.

### C. Proposed Framework and Algorithm

Our proposal is to use adaptive SVR as the regression tool to learn the problem's structure and to enhance the EC search. In the higher dimensional search space, we use SVR to approximate the fitness landscape and analyze the characteristic of the kernel function to predict the region of the global optimum. If this learning mechanism is embedded into a conventional EC algorithm, the EC search capability must be enhanced by the introduction of an adaptive parameter to the SVR learning process. The following EC search framework and algorithm shows the main steps required to implement our proposed evolutionary computation enhancement mechanism by SVR with an OBL-tuned adaptive parameter. The pseudo code for our proposal is shown in Algorithm 1.

### Algorithm 1 Pseudo-code for the Surrogate-Assisted Evolutionary Computation Mechanism by Adaptive SVR with OBL

---

```

1: Initialize EC and randomize adaptive parameter  $P_a$ .
2: for  $G = 1$  to  $maxIter$  do
3:   perform conventional EC operations
4:   approximate fitness landscape by SVR with  $P_a$  and obtain  $Elite$ 
5:   approximate fitness landscape by SVR with opposite  $P_a$  and obtain  $Elite_{op}$ 
6:   if  $f(Elite) > f(Elite_{op})$  then
7:      $FinalElite = Elite_{op}$ 
8:   else
9:      $FinalElite = Elite$ 
10:  end if
11:  Update range of parameter  $P_a$  in accordance with the above decision
12:  if  $FinalEliteFitness > WorstIndividualFitness$  then
13:    replace the worst individual
14:  end if
15: end for
16: return the optimum

```

---

## IV. EXPERIMENTAL EVALUATIONS

### A. Experimental Design and Evaluation Metrics

To investigate the performance of our proposed algorithm, we use differential evolution (DE/1/best/bin) as the optimization method to conduct a series of comparative evaluations. 8 benchmark functions are selected as a benchmark suite [11], Table I shows their range, optimum and characteristic.

The DE and SVR experimental parameters are set as shown in Table II. Experimental evaluations run 30 trials of 1000 generations independently on each of the 8 benchmark functions. All the benchmark functions are set for 10 dimensions. The experimental platform for all evaluations is MATLAB®2009b, running on an Intel(R) Core(TM) i5-2310 CPU, 4G RAM machine under Windows®7. The evaluation is conducted under difficult search conditions; only 50 individuals are used to search the 10-D functions.

TABLE II. DE AND SVR EXPERIMENT PARAMETER SETTINGS

population size	50
scale factor $F$	0.3
crossover rate	0.7
DE operations	DE/best/1/bin
max. search generation, $MAX_{NFC}$	1,000
convergence threshold, $VTR$	$VTR$ in Table V
dimensions of benchmark functions, $D$	10
# of trial runs	30
$P_a$ range	[-1000,1000]

For training the SVR model, the algorithm includes a process for training data selected from the population. In this study, we use the best individual sampling method to select training data [5]. For the proposed algorithm, we set the sampling number at ten. In our experiments, we compare our proposed acceleration approach with the acceleration methods proposed in [2], [12] and a canonical DE, and apply the Wilcoxon signed rank test for the final result of each proposed acceleration method.

We compare the convergence speed of our proposal, DE and other acceleration approaches by measuring the number

TABLE I. 8 BENCHMARK FUNCTIONS USED IN EXPERIMENTAL EVALUATIONS FROM [11] (UNI=UNIMODAL, MUL=MULTIMODAL, SH=SHIFTED, RT=ROTATED, GB=GLOBAL ON BOUNDS, HC=HYBRID COMPOSITION, NM=NUMBER MATRIX, NS=NON-SEPARABLE, AND S=SEPARABLE.)

No.	Name	Form	Range	Optimum	Characters
F1	Sh Sphere	$f(x) = \sum_{i=1}^n x_i^2$	[-100, 100]	-450	Sh-Uni-S
F2	Sh Schwefel 1.2	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i (x_j))^2$	[-100, 100]	-450	Sh-Uni-NS
F3	Sh Rt Elliptic	$f(x) = \sum_{i=1}^n (10^6)^{\frac{i-1}{n-1}} x_i^2$	[-100, 100]	-450	Sh-Rt-Uni-NS
F4	F2 with Noise	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i (x_j))^2 * (1 + 0.4 N(0, 1) )$	[-100, 100]	-450	Sh-Uni-NS
F5	Schwefel 2.6 GB	$f(x) = \max  A_i x - B_i $	[-100, 100]	-310	Uni-NS
F6	Sh Rosenbrock	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	[-100, 100]	390	Sh-Mul-NS
F7	Sh Rt Griewank	$f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}})$	[0, 600]	-180	Sh-Rt-Mul-NS
F8	Sh Rt Scaffer F6	$f(x) = 0.5 + \frac{\sin^2(\sqrt{(x_1^2 + x_2^2)}) - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$	[-100, 100]	-300	Sh-Rt-Mul-NS

of function calls ( $NFCs$ ). A lower NFC means a higher convergence speed. The evaluation is performed with 30 trial runs of 1000 generations, i.e.  $MAX_{NFC} = 1000$ . If an algorithm finds a value less than the value-to-reach ( $VTR$ ) before reaching the maximum number of function calls, it means it has converged. In order to compare convergence speeds, we use the acceleration rate ( $AR$  in Equation (9)) which is defined as follows, based on the  $NFCs$  for the all the algorithms:

$$AR = \frac{NFC_{proposal}}{NFC_{DE}} \quad (9)$$

Where  $AR > 1$  means our proposed method is faster. The number of times for which the algorithm successfully reaches the VTR for each benchmark function is recorded as the success rate ( $SR$  in Equation (10)).

$$SR = \frac{\# \text{ of times after reached } VTR}{\text{total running times}} \quad (10)$$

Further, the average acceleration rate and the average success rate over the benchmark functions are calculated in the final results. A glossary of the abbreviations used in Figure 2, and Table IV are given in Table III.

TABLE III. ABBREVIATIONS USING IN FIGURE 2, AND TABLE IV.

abbreviations	DE which fitness landscape is regressed by
DE	no any method (canonical DE)
DE-SVR	support vector regression
DE-SVR-Adaptive	adaptive support vector regression
DE-OneD-Poly	a two-degree Lagrange interpolation
DE-OneD-Linear	a line power function least square approximation
DE-nD-Poly	a two-degree power function least squares approximation in the original search space with the best sampling [2], [4]

## B. Experimental Results

From Figure 2, and Table IV, we can conclude that our proposal can accelerate EC search significantly for most of the benchmarks. Compared with previous acceleration methods, our proposed methods can obtain better final results in some cases.

TABLE IV. WILCOXON SIGNED-RANK TEST FOR ADAPTIVE SVR WITH ITS COMPETITORS, \*\* AND \* MEAN SIGNIFICANCE OF ( $P < 0.05$ ) AND ( $P < 0.1$ ), RESPECTIVELY. THE ABBREVIATIONS USED HERE ARE DEFINED IN TABLE III.

Methods	F1	F2	F3	F4	F5	F6	F7	F8
DE vs. DE-SVR-Adaptive	**	**	**	**	**	**	**	**
DE-SVR vs. DE-SVR-Adaptive	*							
DE-OneD-Poly vs. DE-SVR-Adaptive	**	**	**	**	**	**	**	
DE-OneD-Linear vs. DE-SVR-Adaptive	**	**	**	**	**	**	**	*
DE-nD-Poly vs. DE-SVR-Adaptive		**		**	**	**		

## V. DISCUSSION

### A. Acceleration Performance of Adaptive SVR with OBL

The main motivation of our proposal is to resolve the complexity of the fitness landscape by projecting the search space into higher dimensional search space by adaptive SVR with OBL. Our proposed EC search framework and algorithm by adaptive SVR can accelerate EC search significantly in most of the benchmark functions. As can be seen in Table IV, our proposal can obtain better final results for some of both the unimodal and multimodal benchmark functions than the other methods. This is thanks to the adaptive parameter tuning, i.e., the construction of different higher dimensional search space. This demonstrates the value of our proposal's originality.

From the AR metric shown in Table V, it can be seen that our proposal outperforms DE-SVR for four of the benchmark functions, and for these benchmark functions the acceleration results are the best of the five proposed methods. From the NFC metric, our proposal does not beat DE-SVR, however, the difference is not great between the two methods.

To investigate its usefulness for accelerating the search process, we conduct a statistical test of our proposal with the competitor methods for all the benchmark functions. Table IV shows the Wilcoxon signed rank test result of each of the proposed methods for each benchmark function in the significant level of 0.05 and 0.1. From the table, for the simple landscape problems, it can be seen that our proposed method can obtain the same acceleration results as SVR, and significantly better than the three other three methods or canonical DE in most of the benchmark tasks.

### B. Comparing Fitness Landscape approximations performed in the Original and Projected One Dimensional Search Space

Comparing among the fitness landscape approximation by fixed SVR, by adaptive SVR, by the regression method in the

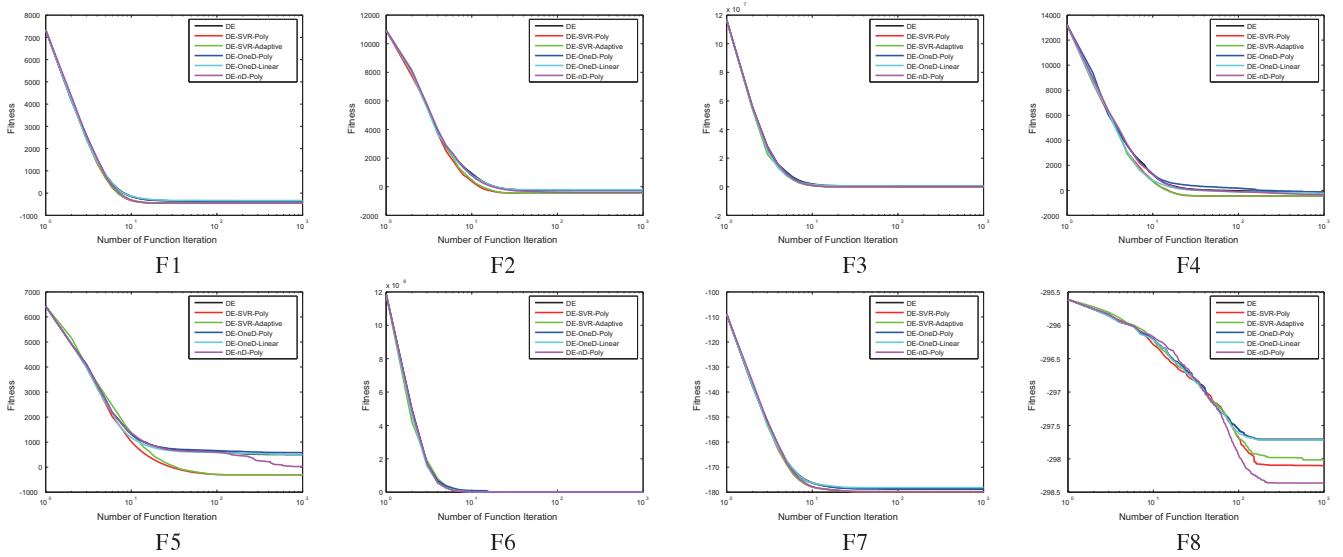


Fig. 2. Average convergence curves of 50 trial runs for F1-F8. There are five acceleration methods for each benchmark function.

original space and in projected lower dimensional search space, the advantages of our proposal are:

1) Rather than the regression in original and projected lower dimensional search space and fixed SVR, we can design and construct any beneficial search space and landscape in a higher dimensional search space by designing a kernel function with our proposed parameter tuning method.

2) In the original or projected lower dimensional search space, the space's topological structure is fixed and cannot be changed, which makes approximation hard for some complex landscapes; the new proposal can change the original search space's topological structure and design the global optimum region to correspond to a certain kernel function by tuning its parameter in a higher dimensional space.

3) It can be obtained better regression performance with a lower sampling number by adaptive SVR and better acceleration performance, another feature of our proposal.

### C. Computational Complexity

For theoretical analysis of the computational complexity of our proposal versus previous fitness landscape approximation methods, we analyze the computational time for each method to make a comparative evaluation. Supposed there is an  $n$  dimensional problem with an observed sample number  $m$ , where we restrict  $m$  such that it is larger than the number of unknown parameters corresponding to the regression model, i.e. a multiple regression model.

For regression in the original space, i.e. the multiple regression model, the regression process includes the following steps: (1) making the normal equation with  $O(n^3)$ , and (2) obtaining the final parameter by the elementary transformation of matrix or adjugate matrix with  $O(n^3)$ . So the total computational complexity is  $O(n^3)$ .

For the regression in the projected one dimensional space, i.e., the unary regression model, the regression process is the

same as that of the multiple regression model. However, the dimension is 1 and its computational complexity depends on the observed sample number, i.e.  $m$ . The computational complexity of the unary regression model is  $O(m)$ . This process will be conducted  $n$  times, the total computational complexity is thus  $O(mn)$ , so when  $m$  approaches  $n$ , it approaches  $O(n^2)$ ; conversely, when  $n$  approaches  $m$ , it approaches  $O(m^2)$ .

For the SVR, the regression process includes the following steps: (1) making the kernel matrix with  $O(m^2)$ , (2) obtain its inverse matrix with  $O(m^3)$ , and (3) computing the parameter, i.e.  $\alpha$  in Equation 5, requiring about  $O(m^3)$ . In total, the process is  $O(m^3)$ ; it depends on  $m$  rather than  $n$ .

From the above analysis, we can conclude that an advantage of SVR is that it requires less computational complexity, because of the computational complexity of SVR depends on the sampling numbers rather than the others which depend on their problem's dimensionality. Additionally, our proposal can adaptively tune the kernel function's parameter.

## VI. CONCLUSION AND FUTURE WORK

We proposed a method to approximate the fitness landscape and accelerate EC search by adaptive SVR using OBL. The method projects the original search space into a higher dimensional search space, such that an approximated landscape can there be obtained, by designing a suitable kernel function with adaptive parameter tuning. We can predict the global optimum location through analyzing the regression model trained by this method. The objective of our proposal is to acquire the related simple landscape for obtaining the global optima information and thus resolving the complexity of landscape in the original search space. From both an empirical evaluation and theoretical analysis, our proposal exhibits better performance both in computational complexity and the final solution for some benchmark tasks.

Kernel function design is a crucial problem in our proposal; it is the kernel which determines the topological structure and

TABLE V. NFC, SR, AR AND VTR (RESPECTIVELY NUMBER OF FUNCTION CALLS, SUCCESS RATIO, ACCELERATION RATIO AND VALUE TO REACH) OF THE 8 BENCHMARK FUNCTIONS USED IN THE EXPERIMENTAL EVALUATIONS. THE NUMBERS EXPRESSED WITH THE BOLD FONT INDICATE BETTER FINAL RESULTS. † INDICATES OUR PROPOSED METHOD CAN OBTAIN BETTER FINAL RESULT THAN DE-SVR.

Methods		F1	F2	F3	F4	F5	F6	F7	F8
DE	NFC	13260	16324	14254	19589	23229	5326	19201	23710
	SR	0.56	0.46	0.52	0.35	0.23	0.82	0.36	0.21
	AR	-	-	-	-	-	-	-	-
DE-SVR	NFC	<b>338</b>	<b>484</b>	<b>346</b>	538	<b>695</b>	<b>272</b>	<b>544</b>	21691
	SR	0.99	0.98	0.99	0.98	0.98	0.99	0.98	0.28
	AR	36.96	<b>33.17</b>	38.7	39.63	<b>37.91</b>	18.58	34.94	2.19
DE-SVR-Adaptive	NFC	341	528	356	<b>536†</b>	823	275	560	22075
	SR	0.99	0.98	0.99	0.98†	0.97	0.99	0.98	0.26
	AR	<b>38.4†</b>	31.57	<b>42.1†</b>	38.97	31.33	<b>19.96†</b>	<b>35.87†</b>	1.83
DE-OneD-Poly	NFC	10486	20234	12514	17165	23282	5337	13434	22780
	SR	0.65	0.33	0.58	0.43	0.22	0.82	0.55	0.24
	AR	3.68	0.86	4.53	10.56	3.42	3.78	8.35	1.44
DE-OneD-Linear	NFC	16231	19293	13404	17243	19444	5290	22161	23710
	SR	0.46	0.36	0.55	0.43	0.35	0.82	0.26	0.21
	AR	8.89	6.39	8.08	12.81	8.29	6.41	6.78	1
DE-nD-Poly	NFC	341	4157	361	12039	12513	287	580	<b>20465</b>
	SR	0.99	0.86	0.99	0.6	0.58	0.99	0.98	0.32
	AR	36.69	11.97	38.02	10.31	4.11	13.71	32.79	<b>7.28</b>
VTR		-401.5955864	-330.7986062	210870.2694	-310.3247581	163.1916265	549888.4994	-179.1783688	-297.936657

characteristic of the higher dimensional space, and regression model construction. There is no kernel function which can regress any landscape with one parameter, so to effectively tune the parameter of the kernel function should be a promising direction for improving this method. We will conduct this work in the future. The sampling method and number is also an important problem in obtaining a better acceleration performance. Our future work will also involve this topic.

#### ACKNOWLEDGMENT

This work was supported in part by Grant-in-Aid for Scientific Research (23500279). Yan Pei would like to thank the Yoshida Scholarship Foundation for its support of his doctoral research.

#### REFERENCES

- [1] Jin Yaochu, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation", *Soft Computing*, Vol.9, No.1, pp.3-12 (2005).
- [2] Pei Y. and Takagi, H., "Accelerating Evolutionary Computation with Elite Obtained in Projected One-Dimensional Spaces", 5th Int. Conf. on Genetic and Evolutionary Computing (ICGEC2011), pp.89-92, Kimmen, Taiwan (Aug. 29 - Sep. 1, 2011).
- [3] Pei Y. and Takagi H., "A Survey on Accelerating Evolutionary Computation Approaches", Third Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR2011), pp.201-206, Dalian, China (14-16, Oct. 2011).
- [4] Pei Y. and Takagi H., "Comparative Evaluations of Evolutionary Computation with Elite Obtained in Reduced Dimensional Spaces", Third Int. Conf. on Intelligent Networking and Collaborative Systems (IN-CoS2011), pp.35-40, Fukuoka, Japan (30, Nov. - 2, Dec. 2011).
- [5] Pei, Y., Zheng S.Q., Tan, Y., and Takagi, H. "An Empirical Study on Influence of Approximation Approaches to Enhance Fireworks Algorithm", 2012 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC2012), pp.1322-1327, Seoul, Korea (Oct. 14-17, 2012).
- [6] Pei.Y. and Takagi, H., "Fourier Analysis of the Fitness Landscape for Evolutionary Search Acceleration", 2012 IEEE Congress on Evolutionary Computation (IEEE CEC 2012), pp.2934-2940, Brisbane, Australia (June, 2012).
- [7] Pei.Y. and Takagi, H., "Comparative Study on Fitness Landscape Approximation with Fourier Transform", Sixth Int. Conf. on Genetic and Evolutionary Computation (ICGEC2012), pp.400-413, Kitakyushu, Japan (25-28, Aug. 2012).
- [8] Pei Y. and Takagi H., "Triple and Quadruple Comparison-Based Interactive Differential Evolution and Differential Evolution", *Foundations of Genetic Algorithms XII (FOGA) 2013 Workshop*, pp.173-182, Adelaide, Australia (16-20 Jan., 2013).
- [9] Pei Yan, "A Chaotic Ergodicity Based Evolutionary Computation Algorithm", The 2013 9th Int. Conf. on Natural Computation (ICNC13), pp.454-469, Shenyang, China (Jul., 23-25, 2013).
- [10] Pei Y. and Takagi H., "Accelerating IEC and EC searches with elite obtained by dimensionality reduction in regression spaces", *Evolutionary Intelligence*, Springer, Vol.6(1), pp.27-40 (2013).
- [11] Suganthan P., Hansen N., Liang J., Deb K., Chen Y., Auger A. and Tiwari S., "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization", Technical Report, Nanyang Technological University, Singapore (2005).
- [12] Takagi H., Ingu T., and Ohnishi K., "Accelerating a GA Convergence by Fitting a Single-Peak Function", *J. of Japan Society for Fuzzy Theory and Intelligent Informatics*, vol.15, no.2, pp.219-229 (2003) (in Japanese).
- [13] Tan Y. and Zhu Y. C., "Fireworks Algorithms for Optimization", *Int. Conf. on Swarm Intelligence (ICSI2010)*, Part II, LNCS 6145, pp.355-364, Beijing, China (12-15, Jun. 2010).
- [14] Tizhoosh, H. R., "Opposition-Based Learning: A New Scheme for Machine Intelligence", *Int. Conf. on Computational Intelligence for Modelling Control and Automation (CIMCA2005)*, vol.I, pp.695-701, Vienna, Austria, (2005).
- [15] Tizhoosh, H. R., "Reinforcement Learning Based on Actions and Opposite Actions", *ICGST Int. Conf. on Artificial Intelligence and Machine Learning (AIML-05)*, Cairo, Egypt, (Dec., 2005).
- [16] Vapnik V., "The Nature of Statistical Learning Theory", Springer, N.Y., (1995).
- [17] Feoktistov V. and Janaqi S., "Hybridization of Differential Evolution with Least-Square Support Vector Machine", *The 13th Annual Belgian-Dutch Conf. on Machine Learning (BENELEARN2004)*, pp.26-31(2004).