Managing Radial Basis Functions for Evolutionary Many-Objective Optimization

Jiangtao Shen
School of Marine Science and Technology
Northwestern Polytechnical University
Xi'an, China
shen_jiang_tao@163.com

Peng Wang
School of Marine Science and Technology
Northwestern Polytechnical University
Xi'an, China
wangpeng305@nwpu.edu.cn

Xinjing Wang
School of Marine Science and Technology
Northwestern Polytechnical University
Xi'an, China
wangxinjing0213@hotmail.com

***Abstract*—This paper proposes a radial basis functions (RBFs) assisted evolutionary algorithm for solving expensive many-objective problems where only a small number of real fitness evaluations are permitted. Two kinds of RBFs are applied in this algorithm, and the differences between the two kinds of RBFs are figured out to provide the estimated errors. By doing this, the estimated individual which has the maximum difference will be evaluated by real functions to strengthen the RBF models. In addition, for each objective, a more suitable RBF is selected for the purpose of making a more accurate approximation of the real functions. The simulation results demonstrate that the proposed algorithm not only performs well on many-objective problems with 10 decision variables, but also shows high efficiency. Besides, the proposed algorithm has good performance on problems with up to 30 decision variables.**

Keywords—Many-objective optimization, evolutionary algorithm, radial basis functions, expensive problem

I. INTRODUCTION

Multi-objective optimization problems (MOPs) appear widely in real world. For MOPs, what we can obtain is a set of trade-off solutions called Pareto Front (PF) rather than a single optimal solution, since the conflicting nature of each objectives. An MOP can be formulated as

$$\begin{aligned} &\min f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})) \\ &s.t.\ \ \mathbf{x} \in \mathbf{X} \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the decision variable. Specifically, for those MOPs with more than three objectives, we called them many-objective problems (MaOPs). A great number of evolutionary algorithms (MOEAs) have been proposed for solving MOPs [1]-[3] and MaOPs [4]-[7], and these algorithms can achieve promising results on MOPs and MaOPs, respectively.

For some engineering problems, it is extremely time-consuming to obtain a real fitness value of an individual, we call them expensive problems, e.g., in computational fluid dynamics (CFD) and finite element analysis (FEA), one simulation may take several hours even several days. However, a large number of real function evaluations are needed before a satisfying solution set is found when MOEAs are employed. Therefore, it is impractical to only use MOEAs to solve expensive MOPs.

One common approach to solve expensive MOPs is introducing surrogate models to approximate the real fitness functions, so that the computation times of real fitness functions can be sharply reduced. Many surrogate-assisted MOEAs have been developed in the past decades, e.g., ParEGO [8], SMSEGO [9], MOEA/D-EGO [10], K-RVEA [11] and CSEA [12]. The first three algorithms apply EGO (also known as Kriging model) and show good performance on 2 and 3-objective problems. K-RVEA is a Kriging-assisted reference vector guided evolutionary algorithm. CSEA is a recently proposed algorithm which employs Feedforward Neural Networks (FNNs) to predict the dominance relationship between candidate solutions. Both K-RVEA and CSEA perform well on MaOPs.

For single objective optimization, two most widely used surrogates are Kriging and RBF [17]. However, most of the current surrogate-assisted MOEAs use Kriging to approximate the real fitness functions and the instances which apply RBFs are very rare. In [20], RBF and NSGA-II are combined to reduce the number of fitness functions evaluations, and it performs well on a set of two-objective problems. However, this method is restricted to handle only multi-objective problems.

Although RBF can't yield mean square errors (MSEs) like Kriging, it is well known by its high modeling efficiency and accuracy. In K-RVEA, the number of training data is fixed since it is time-consuming if the number is excessive. If RBFs are applied, all individuals that evaluated by real functions could be used for constructing RBF models, and the time spent is negligible compared to Kriging models. Therefore, it is expected that the prediction quality of RBFs keeps improving through the optimal process. Moreover, RBFs-assisted RVEA will probably outperform K-RVEA on MOPs with higher dimension due to the property of RBFs. In this paper, an RBFs-assisted reference vector guided MOEA (R-RVEA) is proposed for higher modeling efficiency and quality.

The rest of this paper is organized as follows. Section II introduces the radial basis functions. Section III elaborates

This work was supported in part by the National Natural Science Foundation of China under Grant 51375389, Grant 51875466, and Grant 51805436, and in part by the China Scholarship Council. *(Corresponding author: Peng Wang.)*

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

the proposed algorithm, named R-RVEA. The numerical results of R-RVEA is presents in Section IV. Finally, the conclusions are drawn in Section V.

II. Radial Basis Functions

A Radial basis functions approximates the real objective approximate the real function f as [13]

$$\hat{f}(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\psi} = \sum_{i=1}^{n_c} w_i \psi(\| \mathbf{x} - \mathbf{c}^{(i)} \|) \quad (2)$$

where $\| \mathbf{x}-\mathbf{c}^{(i)} \|$ is the Euclidean distance of an individual $\mathbf{x}$ to the i-th of the n_c basis function centers and $\psi(\bullet)$ represents the basis functions. Several types of the basis functions are listed in TABLE I, where $\| \mathbf{x}-\mathbf{c}^{(i)} \|$ is replaced by r.

TABLE I
Several Kinds of the Radial Basis Functions

linear (L)	$\psi(r) = r$
thin plate spline (TPS)	$\psi(r) = r^2 \ln r$
Gaussian (G)	$\psi(r) = e^{-r^2/(2\sigma^2)}$
multiquadric (MQ)	$\psi(r) = (r^2 + \sigma^2)^{1/2}$
inverse multiquadric (IMQ)	$\psi(r) = (r^2 + \sigma^2)^{-1/2}$

For sampling points X $=\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)},\ldots, \mathbf{x}^{(n)}\}$ and its response $\mathbf{Y} = \{\mathrm{y}^{(1)}, \mathrm{y}^{(2)},\ldots, \mathrm{y}^{(n)}\}$, the basis functions weights $\mathbf{w}$ can be estimated via the interpolation condition

$$\sum_{i=1}^{n_c} w_i \psi(\| \mathbf{x}^{(j)} - \mathbf{c}^{(j)} \|) = \mathrm{y}^{(j)}, \quad j = 1,2,\ldots,n \quad (3)$$

Once the $\mathbf{w}$ is obtained, the objective value of a new individual can be predicted by Eq. (2). In this paper, linear RBF (LRBF) and thin plate spline RBF (TPSRBF) are selected as the surrogate models, since they represent linear and nonlinear models, respectively.

III. The Proposed R-RVEA

A. Framework of R-RVEA

The process of the proposed R-RVEA is presented in algorithm 1. To avoid ambiguity, the objective value of the individuals P is denoted as F, which is a $N\times M$ matrix where N is the number of individuals, M is the number of objectives.

First, initial population P is generated by the Latin hypercube sampling method [14]. An empty archive A is created to store the individuals that have been evaluated by real functions, and the individuals in P are added to A. Matrix C, a $M\times 2$ zero matrix is created to store the scores of each RBF obtained on each objective. Second, two RBFs are trained for each objective using the individuals in A. Then RVEA is applied to explore these RBF models and get the estimated individuals, labeled as P_1 and P_2, which correspond to the linear RBFs and thin plate spline RBFs, respectively. For each objective, more suitable estimated values will be picked out using the count matrix C. After that, the estimation error of each individual is calculated, which

Algorithm 1 R-RVEA

Input: FE_{max}, maximum number of real function evaluations; u, number of individuals to be evaluated and for updating RBF models; g_{max}, prefixed number of generations before updating RBF models;
Output: nondominated solutions of all solutions in A;
$P \leftarrow$ Initialize the population with $11d-1$ individuals using the Latin hypercube sampling;
$FE \leftarrow 11d-1$;
$C \leftarrow M\times 2$ zero matrix;
$A \leftarrow P$;
while $FE \leq FE_{max}$
Train five RBF models for each objective using A;
  $g \leftarrow 0$;
  while $g \leq g_{max}$
    P_1, $P_2 \leftarrow$ Run RVEA with two kinds of RBF models instead of the real functions;
    $g \leftarrow g + 1$;
  end while
  $Es \leftarrow$ Error estimation (P_1, P_2);
  $P \leftarrow$ Select more suitable fitness evaluations for each objective using Algorithm 3(C, P_1, P_2);
  $P_t \leftarrow$ Select infill individuals using Algorithm 4;
  $FE \leftarrow FE + u$;
  $C \leftarrow$ Update the count matrix using Algorithm 2(C, P_t, P_1, P_2);
  $A \leftarrow A \cup P_t$;
end while

Algorithm 2 Updating the count matrix

Input: C, count matrix; P_t, infilled individuals which evaluated by real functions; P_1 and P_2, infilled individuals which estimated by two kinds of RBFs;
Output: Updated C
for $i \leftarrow 1{:}u$
  for $j \leftarrow 1{:}M$
    $k \leftarrow \arg\min_{k=1,\,2} | F_k^{i,j} - F_t^{i,j} |$;
    $C(j, k) \leftarrow C(j, k) + 1$;
  end for
end for

Algorithm 3 Selecting more suitable estimated values for each objective

Input: C, count matrix; P_1 and P_2, individuals which estimated by two kinds of RBFs;
Output: The estimated individuals P
for $i \leftarrow 1{:}M$
  $s \leftarrow \arg\max_{j=1,\,2} C(i, j)$;
  $F(:, i) \leftarrow F_s(:, i)$;
end for

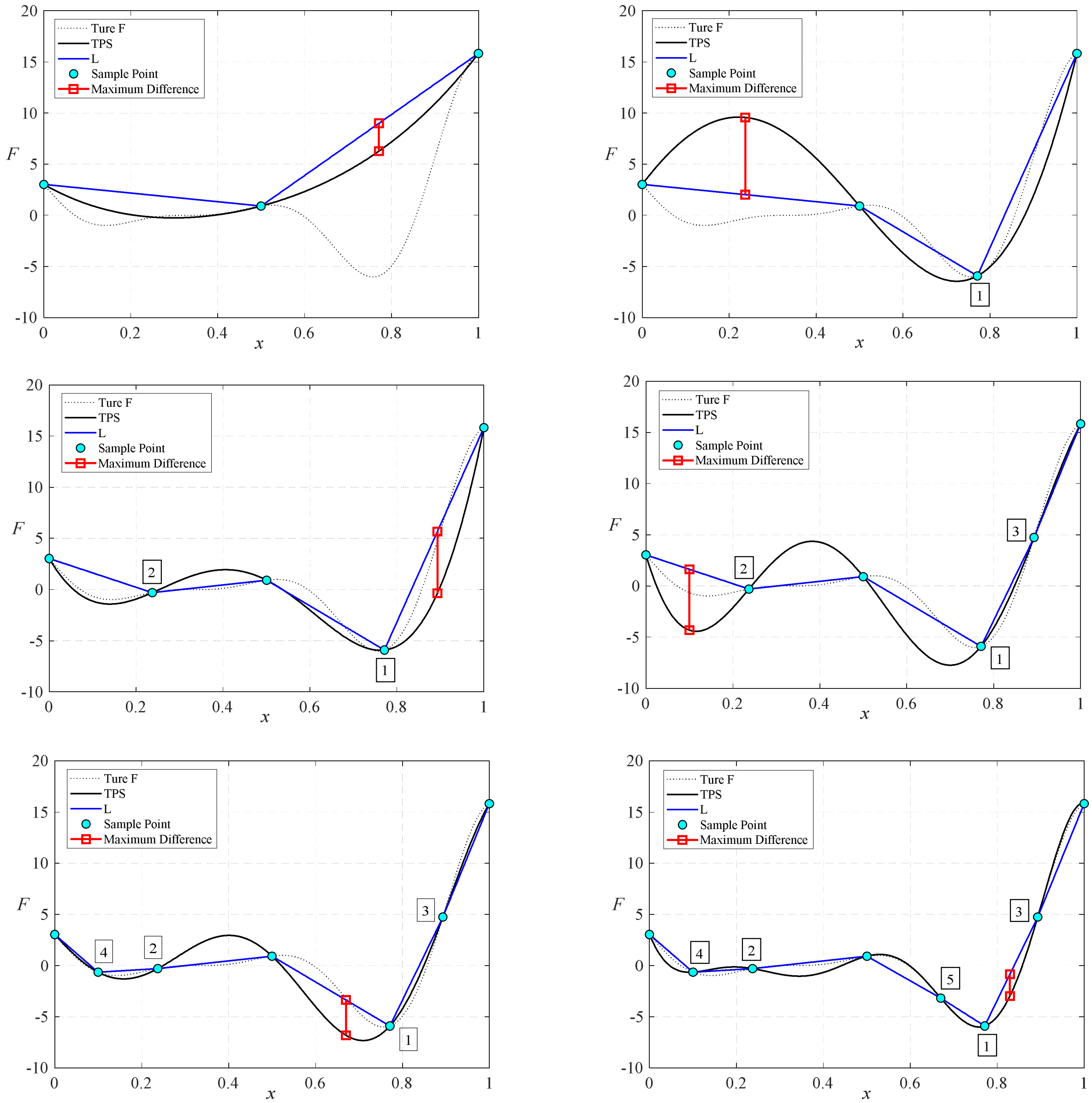

Fig. 1 Illustration of the error estimation strategy.

will be discussed later. Finally, several individuals will be selected and re-evaluated using the real functions and added into A. Then the matrix C will be updated.

In the first iteration, the estimated values of each objective are randomly selected, since the matrix C is a zero matrix. After that, u individuals are selected for updating the RBFs. The new added individuals which have already been evaluated using real functions are denoted as P_t. Then the solutions which are not picked in P_1 and P_2 are deleted. The operation of updating the count matrix C is shown in algorithm 2. Note that $F_k^{i\,j}$ means the estimated value of the j-th objective of the i-th solution by the k-th kind of RBF model.

After the first iteration, the count matrix C will be used for selecting the most suitable estimated values for each objective from two kinds of RBFs. The process is elaborated in the Algorithm 3.

B. Error Estimation

The accuracy of Kriging can be improved through adding samples in the region where the MSE is large. Although RBFs can't generate estimation error like Kriging, the differences between different kinds of RBFs can provide the error information. To illustrate the strategy, LRBF and TPSRBF are used to approximate a one-variable test function $F(x) = (6x-2)^2\sin(12x-4)$. As is shown in Fig. 1, in the first picture, three initial samples are used for the construction of the LRBF and TPSRBF. Then, at each iteration a new point is added in the location where the maximum difference between LRBF and TPSRBF is. After five points added, as can be observed in the last picture, both RBFs become more accurate. Especially the TPS, which has a nearly perfect approximation around the minimum point.

In each iteration, after the final estimated values of two kinds of RBFs are obtained, the estimation errors Es are calculated by Eq. (4), where $|\bullet|$ is the absolute value of each elements of a matrix.

$$Es = |F_1 - F_2| \tag{4}$$

C. Selection of Infilled Individuals

Algorithm 4 Selecting infill individuals

Input: V_a, adaptive reference vectors; V_f, fixed reference vectors; P, final individuals obtained from the RBFs; Ni_{t-1}, number of inactive fixed reference vectors from the previous update; δ, parameter to control convergence and diversity; u, number of individuals to update RBFs;

Output: P_t, u selected individuals;

Dived the active reference vectors into n_u clusters;

Identify the number of empty vectors in the fixed vector set Ni_t;

$\Delta Ni \leftarrow Ni_t - Ni_{t-1}$;

If $\Delta Ni \leq \delta$

    Select one individual from each cluster with minimum APD;

else

    Select one individual from each cluster with maximum estimation error;

end if

In this paper, the infill strategy is the same as that in K-RVEA [11]. In K-RVEA, a set of adaptive vectors V_a and a set of fixed vectors V_f are applied. By comparing the number of empty vectors in the fixed vector set of the previous iteration and of the current iteration, the convergence or diversity which is prior is decided. If the reducing number of empty vectors is more than δ, where $\delta > 0$ is a prefixed small integer, diversity should be prioritized. By contrast, if the reducing number of empty vectors is smaller than δ, priority will be given to the convergence. Then the active adaptive reference vectors are divided into u clusters, from each cluster one individual that has the maximum estimation error (in case diversity is prioritized) or the minimum angle penalized distance (APD) (in case convergence is prioritized) is selected. The process for selecting solutions is shown in Algorithm 4.

D. Differences Between R-RVEA and K-RVEA

Although R-RVEA has the same infill strategy with K-RVEA and both of them apply RVEA to explore the surrogate models, they still have several essential differences which we would like to summarize as follows:

1) K-RVEA uses Kriging models to approximate real objective functions while R-RVEA uses RBF models;

2) In K-RVEA, Kriging can produce MSEs, while the estimation error of individuals in R-RVEA are calculated by the differences between two kinds of RBFs;

3) In K-RVEA, only one kind of Kriging models are used for approximating the real functions while in R-RVEA, two kinds of RBF models are used. Moreover, the most suitable RBF is selected for each objective function for the purpose of a more accurate approximating;

4) In K-RVEA, the number of individuals used for training the Kriging models is fixed, since it is very time-consuming when the number of training data is too large. In R-RVEA, all individuals which have been evaluated by real functions could be used for training the RBF models. Hence, a more accurate approximation can be obtained.

IV. Experimental Studies

A. Performance Indicator

IGD [15] is employed to assess the performance of the algorithms which can be calculated by

$$\mathrm{IGD}(P,R) = \frac{1}{|R|}\sum_{i=1}^{|R|}\min_{1\leq j\leq |P|} d(r_i, p_j) \tag{5}$$

where P is the obtained nondominated solution set and R is a set of evenly distributed reference points on the true Pareto Front, $d(r_i, p_j)$ is the Euclidean distance between p_j in P and r_i in R. $|R|$ is the number of points in R. In this paper, $|R|$ is equal to the closest integer to 10000, which is used for all the test problems. Note that a smaller IGD value represents a better performance. The Wilcoxon rank sum test is adopted at a significance level of 0.05. "+," "−," and "=" indicate that the results by the adaptive strategy is significantly better, significantly worse than and statistically similar to that obtained by R-RVEA, respectively. The results obtained by R-RVEA are highlighted in bold and are shaded if they are the best among the compared algorithms, and are only highlighted in bold if they are the second best among the compared algorithms.

B. Experimental Settings

All test instances are implemented on PlatEMO [16] in MATLAB R2018a and run on a PC with Inter Core i5-8400 with six cores, CPU 2.8GHz and RAM 8.00 GB. The results are executed in normal model, i.e., not in parallel.

1) The population size of all test instances is set to the closest integer to 50.

2) The number of the initial samples is set to $11d-1$ for all test instances, where d is the number of decision variables of the problems.

3) For the problems with 10 decision variables, $FE_{max} = 300$ and $g_{max} = 20$; While for the problems with 30 decision variables, $FE_{max} = 700$ and $g_{max} = 50$.

4) For K-RVEA and R-RVEA, parameter δ is the set to the closest integer to $0.05N$, where N is the population size, and u is set to 5.

5) For CSEA, the number of reference solutions is set to 6, and the nuber of solutions evaluated by surrogate models is set to 3000.

6) Number of independent runs is set to 20 to obtain the statistical results.

7) The simulated binary crossover and polynomial mutation (SBX) are used to generate off-springs in all the considered algorithms. The distribution index is set to 20. The crossover probability is set to 1, the mutation probability is set to $1/d$, where d is the length of decision variable.

TABLE III
IGD VALUES (MEAN AND STANDARD DEVIATION) OF R-RVEA AND ITS TWO VARIANTS ON 3 AND 10-OBJECTIVE DTLZ TEST SUITE WITH 10 DECISION VARIABLES

Problem	M	LR-RVEA	TR-RVEA	R-RVEA
DTLZ1	3	4.9523e+1 (1.25e+1) =	8.0879e+1 (2.13e+1) -	**5.4157e+1 (1.15e+1)**
	10	3.0421e-1 (1.20e-1) =	4.6992e-1 (1.62e-1) -	**2.7486e-1 (7.82e-2)**
DTLZ2	3	7.7082e-2 (6.34e-3) =	7.7082e-2 (3.47e-3) -	**7.3977e-2 (1.84e-3)**
	10	5.1392e-1 (2.40e-2) =	5.2217e-1 (2.06e-2) =	**5.2021e-1 (2.03e-2)**
DTLZ3	3	1.7514e+2 (3.56e+1) =	2.3384e+2 (4.25e+1) -	**1.6589e+2 (4.63e+1)**
	10	1.3498e+0 (4.21e-1) -	1.5620e+0 (5.83e-1) -	**1.0427e+0 (2.79e-1)**
DTLZ4	3	1.8335e-1 (3.48e-2) =	3.3326e-1 (3.12e-2) -	**1.7681e-1 (3.12e-2)**
	10	6.2148e-1 (3.31e-2) =	6.4723e-1 (3.07e-2) -	**6.2729e-1 (2.62e-2)**
DTLZ5	3	3.5868e-2 (4.94e-3) =	3.7299e-2 (3.77e-3) -	**3.2880e-2 (4.74e-3)**
	10	1.3747e-2 (1.99e-3) -	1.2059e-2 (1.69e-3) =	**1.1791e-2 (1.37e-3)**
DTLZ6	3	1.8588e+0 (3.73e-1) =	2.7140e+0 (5.37e-1) -	**1.8362e+0 (2.82e-1)**
	10	5.0946e-2 (2.25e-2) -	7.9907e-2 (2.44e-2) -	**3.7364e-2 (1.62e-2)**
DTLZ7	3	7.0968e-1 (3.90e-1) -	1.2986e-1 (3.79e-2) =	**1.4679e-1 (4.94e-2)**
	10	1.1860e+0 (1.61e-1) -	1.0304e+0 (2.51e-2) =	**1.0279e+0 (3.24e-2)**

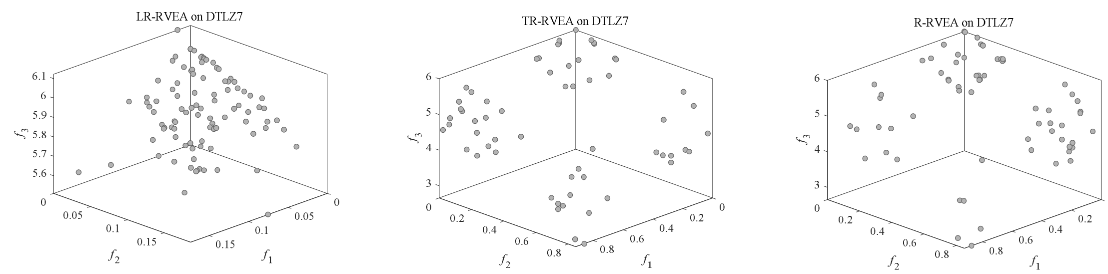

Fig. 2 Nondominated solutions obtained by LR-RVEA, TR-RVEA and R-RVEA on 3-objective DTLZ7 in the run associated with the median IGD value.

C. *Comparing R-RVEA With Its Variants*

TABLE II
NUMBERS OF CALLS TO LRBF AND TPSRBF FOR EACH OBJECTIVE OF THREE-OBJECTIVE DTLZ4 AND DTLZ7 IN A SINGLE RUN

Problems	M	LRBF	TPSRBF
DTLZ4	M_1	146	49
	M_2	140	55
	M_3	58	137
DTLZ7	M_1	4	191
	M_2	8	187
	M_3	93	102

To investigate if the strategy of selecting more suitable estimated values for each objective is effective, R-RVEA is compared with its two variants LR-RVEA and TR-RVEA on DTLZ test suite with 3 and 10 objectives. For LR-RVEA, only LRBFs are applied to approximate real functions, while for TR-RVEA only TPSRBFs are applied. Note that the estimation errors still come from the differences between two kinds of RBF models. The experimental results are presented in TABLE III.

It can be observed from the TABLE III that LR-RVEA performs better on some problems while TR-RVEA performs better on other problems. However, R-RVEA always shows good performance for all the test problems. Fig. 2 presents the nondominated solutions obtained by three kinds of algorithms on 3-objective DTLZ7. It can be observed that LR-RVEA shows poor performance, while both TR-RVEA and R-RVEA perform well on this problem.

In addition, the numbers of calls to LRBF and TPSRBF for each objective of DTLZ4 and DTLZ7 in a single run are shown in TABLE II. As can be observed that for the first two objectives of DTLZ4, it is better to use LRBF to approximate, while for the third objective TPSRBF is better. For the first two objectives of DTLZ7, it is better to use TPSRBF to approximate, for the third objective, LRBF and TPSRBF show same competitiveness. It is easy to understand the contents of the TABLE II, since LR-RVEA show good performance on DTLZ4 problem, TR-RVEA performs well on DTLZ7 problem.

In conclusion, our strategy absorbs the advantages of LRBF and TPSRBF, and is more effective to handle diverse kinds of MOPs.

D. *Comparing R-RVEA With RVEA, K-RVEA and CSEA on Problems With 10 Decision Variables*

The statistical IGD values obtained by R-RVEA and three compared algorithms over 20 independent runs are summarized in TABLE IV. From the 35 test instances, we can find R-RVEA wins in 21 instances in terms of IGD. Obviously, R-RVEA shows the best overall performance

TABLE IV
IGD VALUES (MEAN AND STANDARD DEVIATION) OF RVEA, CSEA, K-RVEA, AND R-RVEA ON 3, 4, 6, 8, AND 10-OBJETIVE DTLZ TEST SUITE WITH 10 DECISION VARIABLES

Problem	*M*	RVEA	CSEA	K-RVEA	R-RVEA
DTLZ1	3	7.3018e+1 (1.54e+1) =	5.9821e+1 (1.52e+1) =	7.8072e+1 (1.24e+1) =	**5.4157e+1 (1.15e+1)**
	4	4.0487e+1 (1.49e+1) =	3.8712e+1 (1.16e+1) =	5.5538e+1 (1.64e+1) =	4.1287e+1 (1.60e+1)
	6	1.6231e+1 (4.74e+0) =	1.9730e+1 (5.52e+0) =	2.7047e+1 (9.39e+0) -	**1.9104e+1 (5.86e+0)**
	8	4.6090e+0 (1.72e+0) =	3.1233e+0 (1.46e+0) =	6.8949e+0 (3.42e+0) =	**4.3283e+0 (1.86e+0)**
	10	5.1677e-1 (3.41e-1) =	2.9478e-1 (8.78e-2) =	4.0490e-1 (1.03e-1) -	**2.7486e-1 (7.82e-2)**
DTLZ2	3	3.0347e-1 (4.59e-2) -	2.3375e-1 (3.14e-2) -	1.2034e-1 (2.45e-2) -	**7.3977e-2 (1.84e-3)**
	4	3.3696e-1 (3.25e-2) -	3.4620e-1 (2.77e-2) -	2.2064e-1 (2.12e-2) -	**1.7752e-1 (7.50e-3)**
	6	4.5553e-1 (3.61e-2) -	4.9045e-1 (3.84e-2) -	3.4031e-1 (2.06e-2) -	**3.0334e-1 (1.09e-2)**
	8	5.8506e-1 (4.05e-2) -	6.1353e-1 (2.19e-2) -	4.5614e-1 (1.50e-2) -	**4.1398e-1 (1.19e-2)**
	10	7.4829e-1 (1.16e-1) -	6.8719e-1 (2.45e-2) -	5.1525e-1 (1.57e-2) =	**5.2021e-1 (2.03e-2)**
DTLZ3	3	2.1747e+2 (4.32e+1) =	1.6626e+2 (2.18e+1) =	2.5333e+2 (4.84e+1) -	**1.6589e+2 (4.63e+1)**
	4	1.5772e+2 (3.46e+1) =	1.0331e+2 (2.05e+1) +	1.8612e+2 (3.95e+1) =	**1.4488e+2 (4.02e+1)**
	6	7.1410e+1 (2.69e+1) =	4.6890e+1 (1.51e+1) +	8.2910e+1 (2.74e+1) =	**6.4153e+1 (1.49e+1)**
	8	1.9460e+1 (8.26e+0) =	1.5404e+1 (7.58e+0) =	1.9936e+1 (8.97e+0) =	2.8036e+1 (1.53e+1)
	10	2.8371e+0 (2.74e+0) -	9.4350e-1 (1.04e-1) =	1.0609e+0 (2.53e-1) =	**1.0427e+0 (2.79e-1)**
DTLZ4	3	4.5065e-1 (1.64e-1) -	4.0823e-1 (1.34e-1) -	2.8928e-1 (1.26e-1) -	**1.7681e-1 (3.12e-2)**
	4	5.1603e-1 (1.96e-1) -	3.0570e-1 (3.39e-2) =	4.2195e-1 (9.68e-2) -	**3.0695e-1 (3.58e-2)**
	6	6.4316e-1 (1.18e-1) -	5.1277e-1 (3.18e-2) =	4.8137e-1 (3.53e-2) =	**4.8402e-1 (4.66e-2)**
	8	7.2857e-1 (7.04e-2) -	5.9399e-1 (5.83e-2) =	5.4671e-1 (2.52e-2) =	**5.7409e-1 (5.80e-2)**
	10	9.4279e-1 (1.03e-1) -	6.2825e-1 (4.56e-2) =	6.5640e-1 (2.10e-2) -	**6.2729e-1 (2.62e-2)**
DTLZ5	3	2.1267e-1 (2.99e-2) -	1.1895e-1 (2.58e-2) -	6.2930e-2 (9.20e-3) -	**3.2880e-2 (4.74e-3)**
	4	2.2867e-1 (5.74e-2) -	1.1730e-1 (3.06e-2) -	5.5316e-2 (1.92e-2) -	**3.3405e-2 (9.20e-3)**
	6	2.6340e-1 (6.88e-2) -	8.6218e-2 (1.88e-2) -	3.4934e-2 (1.03e-2) =	**3.3651e-2 (5.74e-3)**
	8	1.5925e-1 (3.58e-2) -	4.8502e-2 (1.04e-2) -	2.5174e-2 (4.72e-3) =	**2.2673e-2 (6.81e-3)**
	10	2.1478e-1 (1.76e-1) -	1.4814e-2 (1.25e-3) -	1.1855e-2 (1.38e-3) =	**1.1791e-2 (1.37e-3)**
DTLZ6	3	6.2057e+0 (2.85e-1) -	5.1020e+0 (5.19e-1) -	2.7004e+0 (4.43e-1) -	**1.8362e+0 (2.82e-1)**
	4	5.0263e+0 (4.16e-1) -	5.1091e+0 (4.65e-1) -	2.2131e+0 (6.59e-1) -	**1.0748e+0 (6.32e-1)**
	6	3.4126e+0 (3.93e-1) -	3.2461e+0 (4.75e-1) -	1.2628e+0 (3.75e-1) -	**7.8128e-1 (2.76e-1)**
	8	2.1033e+0 (3.21e-1) -	1.4708e+0 (3.63e-1) -	3.7960e-1 (1.19e-1) -	**2.5324e-1 (1.60e-1)**
	10	4.7032e-1 (2.70e-1) -	9.6364e-2 (5.45e-2) -	8.9066e-2 (4.09e-2) -	**3.7364e-2 (1.62e-2)**
DTLZ7	3	2.8562e+0 (1.07e+0) -	1.7753e+0 (5.58e-1) -	1.0679e-1 (7.32e-3) +	**1.4679e-1 (4.94e-2)**
	4	3.2357e+0 (1.29e+0) -	3.0797e+0 (1.04e+0) -	3.0280e-1 (1.78e-1) =	**2.9659e-1 (4.88e-2)**
	6	3.6262e+0 (9.57e-1) -	7.7863e+0 (3.08e+0) -	5.5085e-1 (7.47e-2) =	**5.9219e-1 (1.14e-1)**
	8	3.9248e+0 (2.22e+0) -	6.3717e+0 (3.46e+0) -	8.8887e-1 (6.75e-2) =	**8.8686e-1 (3.74e-2)**
	10	1.9306e+0 (5.32e-1) -	2.2519e+0 (4.85e-1) -	1.0381e+0 (5.09e-2) =	**1.0279e+0 (3.24e-2)**

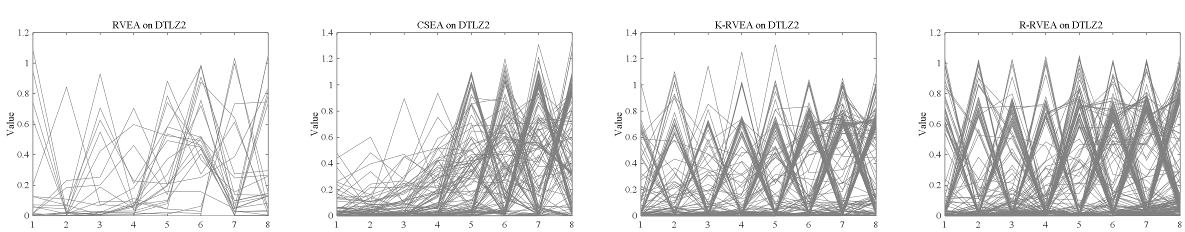

Fig. 3 Nondominated solutions obtained by RVEA, CSEA, K-RVEA, and R-RVEA on 8-objective DTLZ2 in the run associated with the median IGD value.

among the four compared algorithms.

DTLZ1 and DTLZ3 are two multimodal problems, and it is difficult to obtain a set of solutions which show good convergence. As can be observed from TABLE IV that R-RVEA and CSEA show the best performance on all the DTLZ1 and DTLZ3 instances. For DTLZ2, R-RVEA

TABLE V
IGD VALUES AND RUNTIME (MEAN AND STANDARD DEVIATION) OF CSEA, K-RVEA, AND R-RVEA ON 3-OBJECTIVE DTLZ TEST SUITE WITH 30 DECISION VARIABLES

Problem	CSEA		K-RVEA		R-RVEA	
	IGD	Runtime/sec	IGD	Runtime/sec	IGD	Runtime/sec
DTLZ1	2.9078e+2 (3.81e+1) =	6.8124e+2 (4.10e+2) -	4.9957e+2 (3.60e+1) -	1.3138e+3 (3.83e+1) -	**3.0353e+2 (6.12e+1)**	**1.4301e+2 (5.49e+0)**
DTLZ2	4.7147e-1 (8.65e-2) -	6.4518e+2 (2.39e+2) -	5.5942e-1 (7.90e-2) -	1.3077e+3 (5.18e+1) -	**8.5168e-2 (3.70e-3)**	**1.2819e+2 (3.21e+0)**
DTLZ3	8.1895e+2 (1.54e+2) =	1.8588e+3 (9.93e+2) -	1.4213e+3 (7.85e+1) -	1.3088e+3 (2.31e+1) -	**8.4462e+2 (1.13e+2)**	**1.3495e+2 (2.98e+0)**
DTLZ4	4.4581e-1 (7.77e-2) -	1.0374e+3 (6.99e+2) -	1.0720e+0 (1.01e-1) -	1.2306e+3 (3.18e+1) -	**2.8902e-1 (1.03e-1)**	**1.3773e+2 (3.77e+0)**
DTLZ5	4.4378e-1 (1.03e-1) -	8.3556e+2 (4.15e+2) -	4.1017e-1 (5.23e-2) -	1.2167e+3 (2.77e+1) -	**8.3750e-2 (1.06e-2)**	**9.4669e+1 (2.56e+0)**
DTLZ6	2.0826e+1 (1.02e+0) -	8.2702e+1 (2.52e+1) +	1.1651e+1 (7.19e-1) -	1.3094e+3 (2.40e+1) -	**1.0269e+1 (1.37e+0)**	**1.3037e+2 (3.62e+0)**
DTLZ7	2.1929e+0 (8.44e-1) -	6.4972e+2 (4.41e+2) -	9.7470e-2 (8.48e-3) =	1.0703e+3 (5.05e+1) -	**9.8775e-2 (4.23e-2)**	**1.2834e+2 (6.30e+0)**

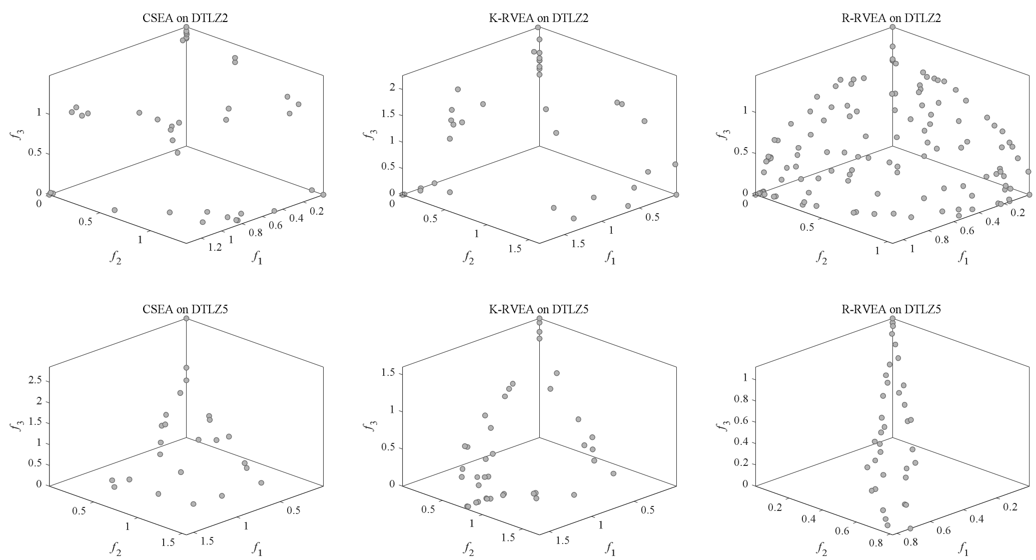

Fig. 4 Nondominated solutions obtained by CSEA, K-RVEA, and R-RVEA on 3-objective DTLZ2 and DTLZ5 with 30 decision variables in the run associated with the median IGD value.

achieves the best performance. Fig. 3 presents the nondominated solutions obtained by four kinds of algorithms on 8-objective DTLZ2 problem [18] [19]. It can be observed that R-RVEA outperforms other algorithms in convergence and diversity on this test instance. For DTLZ4, both R-RVEA and CSEA show the best performance. DTLZ5 and DTLZ6 are two problems whose PFs are degenerated, it can be observed that R-RVEA shows the best performance on these test instances. For DTLZ7, R-RVEA and K-RVEA show similar performance, the performance of RVEA and CSEA are not better than them.

E. Comparing R-RVEA With K-RVEA and CSEA on Problems With 30 Decision Variables

Kriging-assisted MOEAs have mainly been tested on problems with 10 decision variables, since it is time-consuming to train Kriging models when the number of decision variables increases. However, RBF models are well known for its high efficiency, which are expected to address MOPs with more than 10 decision variables.

The statistical IGD values and runtime obtained by R-RVEA and two compared algorithms over 20 independent runs are summarized in TABLE V. It can be observed that R-RVEA achieves the best overall performance when considering no matter IGD values or runtime. Nondominated solutions obtained by each algorithm on 3-objective DTLZ2 and DTLZ5 with 30 decision variables are plotted in Fig. 4. It can be seen that only R-RVEA successfully approximates the PFs of DTLZ2 and DTLZ5. In summary, R-RVEA is efficient to tackle MOPs with up to 30 decision variables.

V. CONCLUSION

In this paper, an RBFs-assisted MOEA, namely R-RVEA is proposed for solving expensive MOPs. In R-RVEA, two kinds of RBF models are applied to provide the error information. In addition, for each objective, more suitable estimated values are selected according to a count matrix for the purpose of more accurate approximation.

In order to verify the effectiveness of the selecting strategy, a group of contrast experiments is conducted. To investigate the performance of R-RVEA, it has been compared with RVEA, K-RVEA and CSEA on DTLZ test suite with 10 decision variables. Moreover, we also compare R-RVEA with K-RVEA and CSEA on 3-objective DTLZ test suite with 30 decision variables to demonstrate the efficiency of RBFs assisted MOEA. The experiment results indicate that R-RVEA is highly competitive compared with the state-of-the-art surrogate-assisted MOEAs for expensive MOPs.

This paper demonstrates that managing RBFs for solving expensive MOPs is effective. However, it still has much room for improvement. For example, the performance of R-RVEA remains to be verified on real world expensive MOPs. In addition, RBFs assisted-MOEAs are expected to tackle middle even high-dimension expensive MOPs.

REFERENCE

[1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.

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

[3] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun and J. Wu, "MOEA/D with Adaptive Weight Adjustment," in *Evolutionary Computation*, vol. 22, no. 2, pp. 231-264, June 2014.

[4] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints," in *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577-601, Aug. 2014.

[5] K. Li, K. Deb, Q. Zhang and S. Kwong, "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition," in *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694-716, Oct. 2015.

[6] R. Cheng, Y. Jin, M. Olhofer and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773-791, Oct. 2016.

[7] H. Wang, L. Jiao and X. Yao, "Two_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524-541, Aug. 2015.

[8] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," in *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50-66, Feb. 2006.

[9] Ponweiser, Wolfgang, et al. "Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted S-Metric Selection." *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, 2008.

[10] Q. Zhang, W. Liu, E. Tsang and B. Virginas, "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model," in *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456-474, June 2010.

[11] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen and K. Sindhya, "A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129-142, Feb. 2018.

[12] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang and Y. Jin, "A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74-88, Feb. 2019.

[13] Forrester, Alexander, Andras Sobester, and Andy Keane. Engineering design via surrogate modelling: a practical guide. John Wiley & Sons, 2008, pp. 45-49.

[14] McKay, Michael D., Richard J. Beckman, and William J. Conover. "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code." *Technometrics* 42.1 (2000): 55-61.

[15] Q. Zhang, A. Zhou and Y. Jin, "RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm," in *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41-63, Feb. 2008.

[16] Y. Tian, R. Cheng, X. Zhang and Y. Jin, "PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]," in *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73-87, Nov. 2017.

[17] X. Wang, G. G. Wang, B. Song, P. Wang and Y. Wang, "A Novel Evolutionary Sampling Assisted Optimization Method for High-Dimensional Expensive Problems," in *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 815-827, Oct. 2019.

[18] M. Li, L. Zhen and X. Yao, "How to Read Many-Objective Solution Sets in Parallel Coordinates [Educational Forum]," in *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 88-100, Nov. 2017.

[19] Zhen, Liangli, et al. "Adjusting Parallel Coordinates for Investigating Multi-objective Search." in book Simulated Evolution and Learning: 11th International Conference, SEAL 2017, Shenzhen, China, November 10–13, 2017, pp.224-235.

[20] L. V. Santana-Quintero, V. A. Serrano-Hernandez, C. A. C. Coello, A. G. Hernandez-Diaz and J. Molina, "Use of Radial Basis Functions and Rough Sets for Evolutionary Multi-Objective Optimization," *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, Honolulu, HI, 2007, pp. 107-114.