Combining Surrogate Models and Local Search for Dealing with Expensive Multi-objective Optimization Problems

Saúl Zapotecas Martínez
CINVESTAV-IPN,
Departamento de Computación
Evolutionary Computation Group
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO,
email: `saul.zapotecas@gmail.com`

Carlos A. Coello Coello
CINVESTAV-IPN,
Departamento de Computación
Evolutionary Computation Group
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO,
email: `ccoello@cs.cinvestav.mx`

***Abstract*—The development of multi-objective evolutionary algorithms (MOEAs) assisted by surrogate models has significantly increased in the last few years. However, in real-world applications, the high modality and dimensionality that functions normally have, often causes problems to such models. Therefore, if the Pareto optimal set of a multi-objective optimization problem is located in a search space in which the surrogate model is not able to shape the corresponding region, the search could be misinformed and thus converge to wrong regions. This has naturally motivated the idea of incorporating refinement mechanisms to such approaches. In this paper, we present a local search mechanism which improves the search of a MOEA assisted by surrogate models. Our preliminary results indicate that our proposed approach can produce good quality results when it is restricted to performing only between 1,000 and 5,000 fitness function evaluations. Our proposed approach is validated using a set of standard test problems and an airfoil design problem.**

I. INTRODUCTION

Multi-objective evolutionary algorithms (MOEAs) have been successfully adopted to solve multi-objective optimization problems (MOPs) in a wide variety of engineering and scientific problems [1]. However, in real-world applications, it is common to find objective functions which are very expensive to evaluate (in terms of computational time). This has considerably limited the use of MOEAs to deal with these types of problems. In recent years, several researchers have developed different strategies for reducing the computational time (measured in terms of the number of fitness function evaluations) that a MOEA requires to solve a determined problem. The use of surrogate models has been one of the most commonly adopted techniques to solve computationally expensive problems. In the last few years, several authors have reported the use of surrogate models for dealing with MOPs, see for example [5], [9], [10], [16], [25]. The main idea behind these approaches is to explore the regions that, according to the surrogate model, promise suitable compromises among the objectives. However, the prediction error of such models often directs the search towards regions in which, no Pareto optimal solutions are found. Because of this, an important number of researchers have tried to improve the prediction of surrogate models by adding new solutions to the training set and then retraining the surrogate model at each iteration of the MOEA, see for example [25], [20].

Recently, in [23], a MOEA based on decomposition assisted by Radial Basis Functions (RBFs) was introduced. This evolutionary approach (called MOEA/D-RBF) employs different kernels for building different RBF networks. Each RBF network provides different shapes of the search space and all of them provide information to predict the value of an arbitrary solution. In fact, the high modality and dimensionality of some problems, often constitute major obstacles for surrogate models. Therefore, if MOEA/D-RBF is not able to shape the region in which the Pareto set is contained, the search could be misinformed and converge to wrong regions. This has motivated the idea of incorporating procedures to refine the solutions provided by surrogate models, such as local search mechanisms. In general, the use of local search mechanisms combined with MOEAs assisted by surrogate models has been only scarcely explored in the specialized literature.

In 2009, Georgopoulou and Giannakoglou [6] proposed a multi-objective memetic algorithm assisted by RBFs. The local search mechanism uses a function which corresponds to an ascent method that incorporates gradient values provided by the surrogate model. In order to improve the function prediction, at each iteration, the model is retrained by considering the current offspring, parent and elite populations. A more recent work, is the one presented by Zapotecas and Coello [20]. The proposed multi-objective memetic algorithm is assisted by Support Vector Regression (SVR). The local search mechanism is directed by several scalarization functions, which are solved by using the Hooke and Jeeves algorithm [8]. The local search is assisted by the surrogate model and the improved solutions are incorporated into the current population of the MOEA by using Pareto ranking. To improve the model accuracy, the surrogate model is retrained by adding the new solutions found to the training set.

The two above approaches assist their local search mechanisms with surrogate models. Therefore, even though these approaches use refinement mechanisms, the prediction error may misguide the local search engine. In this paper,

---

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN.

The second author gratefully acknowledges support from CONACyT project no. 103570.

we present a MOEA assisted by surrogate models adopting a local search engine (which is not assisted by surrogate models) in order to improve the search. The proposed approach is based on MOEA/D-RBF [23] and the refinement mechanism is based on Nelder and Mead's method [15]. We hypothesized that an appropriate combination of the explorative power of MOEA/D-RBF with the exploitative power of a local search engine, could improve the performance of MOEA/D-RBF when performing a low number of fitness function evaluations. To validate or proposed approach, we adopt standard test functions and a real-world problem.

The remainder of this paper is organized as follows. In Section II, we present the basic concepts required to understand the rest of the paper. Section III presents the general framework of MOEA/D-RBF, which is adopted in our study. In Section IV, we describe in detail our proposed approach. In Section V, the test problems adopted to validate our approach are described. In Section VI, we show and discuss the results obtained by our proposed approach. Finally, in Section VII, we provide our conclusions and some possible paths for future research.

II. BASIC CONCEPTS

A. *Multi-Objective Optimization*

Without loss of generality we will assume only minimization problems. Thus, a nonlinear multi-objective optimization problem can be formulated as:

$$\min_{\mathbf{x}\in\Omega} \quad \mathbf{F}(\mathbf{x}) \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ defines the decision variable space and $\mathbf{F} : \Omega \rightarrow \mathbb{R}^k$ defines the vector of objective functions $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$, such that $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear function. In order to describe the concept of optimality in which we are interested on, the following definitions are introduced [14].

Definition Let's assume that $\mathbf{x}, \mathbf{y} \in \Omega$, then, we say that $\mathbf{x}$ *dominates* $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) if and only if, $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\mathbf{F}(\mathbf{x}) \neq \mathbf{F}(\mathbf{y})$.

Definition Let $\mathbf{x}^\star \in \Omega$, we say that $\mathbf{x}^\star$ is a *Pareto optimal* solution, if there is no other solution $\mathbf{y} \in \Omega$ such that $\mathbf{y} \prec \mathbf{x}^\star$.

Definition The *Pareto optimal set* PS is defined by: $PS = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal solution}\}$, and the *Pareto optimal front* PF is defined as: $PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in PS\}$.

We are interested in maximizing the number of elements of the Pareto optimal set and maintaining a well-distributed set of solutions along the Pareto optimal front.

B. *Decomposing Multi-Objective Optimization Problems*

In the specialized literature, there are several approaches for transforming a MOP into a single-objective optimization subproblem [4], [14]. These approaches use a weighted vector as their search direction. In this way, and under certain assumptions (e.g., the minimum is unique, the weighting coefficients are positive, etc.), a Pareto optimal point is achieved by solving such subproblems. Therefore, an approximation of the Pareto optimal front can be achieved by decomposing a MOP into several single-objective optimization problems. Among these methods, perhaps the two most widely used are the *Tchebycheff* and the *Weighted Sum* approaches. It is worth noting, however, that the approaches based on boundary intersection have certain advantages over those based on either Tchebycheff or the Weighted Sum [2], [24]. In the following, we briefly describe a method based, precisely, on the boundary intersection approach, which is referred to in this work.

1) Penalty Boundary Intersection Approach: The Penalty Boundary Intersection (PBI) approach proposed by Zhang and Li [24], uses a weighted vector $\mathbf{w}$ and a penalty value θ for minimizing both the distance to the utopian vector d_1 and the direction error to the weighted vector d_2 from the solution $\mathbf{F}(\mathbf{x})$. Therefore, the optimization problem can be stated as:

$$\text{Minimize:} \quad g(\mathbf{x}|\mathbf{w}, \mathbf{z}^\star) = d_1 + \theta d_2 \tag{2}$$

where,

$$d_1 = \frac{||(\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star)^T \mathbf{w}||}{||\mathbf{w}||} \quad \text{and} \quad d_2 = \left|\left|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^\star) - d_1 \frac{\mathbf{w}}{||\mathbf{w}||}\right|\right|$$

such that: $\mathbf{x} \in \Omega$ and $\mathbf{z}^\star = (z_1, \ldots, z_k)^T$, where $z_i = \min\{f_i(\mathbf{x}) | \mathbf{x} \in \Omega\}$.

In this way, a good representation of the Pareto front can be generated by solving a set of problems defined by a well-distributed set of weighted vectors. That has been the main incentive for the development of current decomposition-based MOEAs, see for example [24], [17], [21].

III. THE FRAMEWORK OF MOEA/D-RBF

The multi-objective evolutionary algorithm based on decomposition assisted by RBF networks (MOEA/D-RBF) proposed in [23] decomposes the problem (1) into N single-objective optimization problems. MOEA/D-RBF uses a well-distributed set of N weight vectors $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ to define a set of single-objective optimization subproblems by using the PBI approach. Each subproblem is solved by using the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [24], which is assisted by the RBF networks. Algorithm 1 shows the general framework of MOEA/D-RBF. In the following sections, we briefly describe the components of MOEA/D-RBF outlined in Algorithm 1.

A. *Building the surrogate model*

MOEA/D-RBF uses three different kernels (Gaussian, multi-quadratic and inverse multi-quadratic) for building different RBF networks. Each RBF network provides different shapes of the search space and all of them provide information to predict the value of an arbitrary solution. Considering $T_{set} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_t}\}$ as the set of N_t solutions evaluated with the real fitness function. The RBF networks are trained by using the set T_{set} and adopting different kernels for each one.

Once the three RBF networks are built, the prediction of the function is carried out. Let $\varphi_{GK}(\mathbf{x}), \varphi_{MK}(\mathbf{x})$ and $\varphi_{IMK}(\mathbf{x})$ be the predicted value given by RBF networks using the Gaussian, multi-quadratic and inverse multi-quadratic kernel, respectively. These three RBF networks cooperate by providing information of the search space that

Algorithm 1: General framework of MOEA/D-RBF

```
Input:
  $W = \{\mathbf{w}_i, \ldots, \mathbf{w}_N\}$: A well-distributed set of weight vectors.
  $N_t$: The number of points in the initial training set.
  $E_{max}$: The maximum number of evaluations allowed in MOEA/D-RBF.
Output:
  $A$: An approximation to the $PF$.
begin
    INITIALIZATION: Generate a set $T_{set} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_t}\}$ of $N_t$ points such that $\mathbf{x}_i \in \Omega$ $(i = 1, \ldots N_t)$, by using an experimental design method. Evaluate the $\mathbf{F}$-functions values of these points. Set $A$ as the set of nondominated solutions found in $T_{set}$. Set $n_{eval} = N_t$. Generate a population $\hat{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of $N$ individuals such that $\mathbf{x}_i \in \Omega$ $(i = 1, \ldots N)$, by using an experimental design method. $stopping\_criterion = FALSE$.
    while ($stopping\_criterion == FALSE$) do
        MODEL BUILDING: Using the $\mathbf{F}$-function values of the points in $T_{set}$, build the predictive surrogate model by using different RBF networks. Calculate the weights for each RBF network according to its training error in $T_{set}$, see Section III-A.
        EVALUATE $\hat{P}$: Evaluate the population $\hat{P}$ using the surrogate model.
        FIND AN APPROXIMATION TO $PF$: By using MOEA/D, the surrogate model and the population $\hat{P}$, obtain $\hat{P}^\star = \{\hat{\mathbf{x}}_i, \ldots, \hat{\mathbf{x}}_{N_t}\}$, where $\hat{P}^\star$ is an approximation to $PF$.
        SELECT POINTS FOR UPDATING $T_{set}$: By using the selection scheme, select a set of solutions from $\hat{P}^\star$ to be evaluated and included in the training set $T_{set}$. Update $A$ using the selected solutions. For each evaluated solution, set $n_{eval} = n_{eval} + 1$. If $n_{eval} < E_{max}$ then $stopping\_criteria = TRUE$. For a detailed description of this step see Section III-B.
        UPDATE POPULATION $\hat{P}$: Update the population $\hat{P}$ according to the updating scheme, see Section III-C.
    end
    return $A$;
end
```

they model. Therefore, the function prediction $\hat{f}$ for an arbitrary $\mathbf{x} \in \Omega$ is defined by:

$$\hat{f}(\mathbf{x}) = \lambda_1 \cdot \varphi_{GK}(\mathbf{x}) + \lambda_2 \cdot \varphi_{MK}(\mathbf{x}) + \lambda_3 \cdot \varphi_{IMK}(\mathbf{x}) \tag{3}$$

where $\mathbf{\Lambda} = (\lambda_1, \lambda_2, \lambda_3)^T$ is a weight vector, i.e. $\lambda_i \geq 0$ and $\sum_{i=1}^{3} \lambda_i = 1$. The weight vector $\mathbf{\Lambda}$ is then calculated as $\lambda_i = \frac{\alpha_i}{|T_{set}|}, i = 1, 2, 3$, where α_i is the number of solutions in T_{set} with the lowest prediction error for the i^{th} RBF network (Gaussian, multi-quadratic and inverse multi-quadratic, respectively).

B. Selecting Points to Evaluate

Let W be the well-distributed set of weight vectors used by MOEA/D. Let $\hat{P}^\star$ be the approximation to PF obtained by MOEA/D. Let W_s be a well-distributed set of weight vectors, such that $|W_s| < |W|$. For each $\mathbf{w}_i^s \in W_s$, a neighborhood $B_s(\mathbf{w}_i^s) = \{\mathbf{w}_1, \ldots, \mathbf{w}_{N_a}\}$ is defined, such that $\mathbf{w}_1, \ldots, \mathbf{w}_{N_a} \in W$ are the $N_a = \lfloor \frac{N}{N_s} \rfloor$ closest weight vectors from W to $\mathbf{w}_i^s$. Once the neighborhoods $B_s(\mathbf{w}_i^s)$ have been defined, a set of solutions is selected to be included in the training set T_{set}, according to the next description.

1) Selecting Points to be Evaluated using the Real Fitness Function: MOEA/D-RBF takes from $\hat{P}$ a set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_s}\}$ of N_s solutions to be evaluated using the real fitness function. Each solution in S is selected such that it minimizes the problem defined by a weight vector $\mathbf{w}_j \in B_s(\mathbf{w}_i^s)$, where $i = 1, \ldots, N_s$ and $j = 1, \ldots N_a$.

At each call of the selection procedure, the weight vector $\mathbf{w}_j$ is selected by sweeping the set of weight vectors in $B_s(\mathbf{w}_i^s)$ in a cyclic way. No solution in S should be duplicated. If this is the case, the repeated solution should be removed from S. For each newly evaluated solution, n_{eval} is increased in one, if $n_{eval} \geq E_{max}$ then set $stopping\_criterion = TRUE$, where n_{eval} and E_{max} are the current and the maximum number of fitness function evaluations, respectively.

2) Updating the Training Set and the External Archive: The maximum number of solutions in the training set T_{set} is defined by the parameter N_t. The updating of T_{set} is carried out by defining a well-distributed set W_t of N_t weight vectors. Therefore, the best N_t different solutions from $T = \{T_{set} \cup S\}$, such that they minimize the subproblems defined by each weight vector $\mathbf{w}_i^t \in W_t$ $(i = 1, \ldots, N_t)$, are used to update T_{set}. If, after updating the training set, any solution $\mathbf{s}_j \in S$ was not selected to be included in T_{set}, then, it is added by replacing the closest solution (in the objective space) in T_s. With this, all solutions in S are included in T_{set} and the model can be improved even if it has been previously misinformed.

The external archive A contains the nodominated solutions found during the search. For each $\mathbf{s}_j \in S$, the external archive is updated by removing from A all the solutions dominated by $\mathbf{s}_j$, and then, $\mathbf{s}_j$ is stored in A if no solutions in A dominate $\mathbf{s}_j$.

C. Updating the Population

Once the external archive is updated, the population $\hat{P}$ is also updated for the next iteration of MOEA/D. Considering the external archive A as the set of nondominated solutions found by MOEA/D-RBF, the population $\hat{P}$ of N solutions is updated according to the following description.

Let $\mathbf{m}$ and σ be the average and standard deviation of the solutions contained in A. Then, new bounds in the search space are defined according to:

$$\begin{array}{rcl} \mathbf{L}_{bound} & = & \mathbf{m} - \sigma \\ \mathbf{U}_{bound} & = & \mathbf{m} + \sigma \end{array}$$

where $\mathbf{L}_{bound}$ and $\mathbf{U}_{bound}$ are the vectors which define the lower and upper bounds of the new search space, respectively.

Once the new bounds have been defined, a set Q of $N - |A|$ solutions is generated by means of the Latin hypercube sampling method [13] in the new search space. The population $\hat{P}$ is then redefined by the union of Q and A, that is $\hat{P} = \{Q \cup A\}$.

In this section, we have briefly explained the components of MOEA/D-RBF. However, in order for a more detailed description of MOEA/D-RBF, see [23].

Algorithm 2: General framework of MOEA/D-RBF+LS

```
Input:
$W = \{\mathbf{w}_i, \ldots, \mathbf{w}_N\}$: A well-distributed set of weight vectors.
$N_t$: The number of points in the initial training set.
$E_{max}$: The maximum number of evaluations allowed in MOEA/D-RBF+LS.
Output:
$A$: An approximation to the $PF$.
begin
    Step 1. INITIALIZATION: Generate a set $T_{set} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_t}\}$ of $N_t$ points such that $\mathbf{x}_i \in \Omega$ ($i = 1, \ldots N_t$), by using an experimental design method. Evaluate the $\mathbf{F}$-functions values of these points. Set $A$ as the set of nondominated solutions found in $T_{set}$. Set $n_{eval} = N_t$. Generate a population $\hat{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of $N$ individuals such that $\mathbf{x}_i \in \Omega$ ($i = 1, \ldots N$), by using an experimental design method. $stopping\_criterion = FALSE$.
    while ($stopping\_criterion == FALSE$) do
        Step 2. MODEL BUILDING: See section III-A.
        Step 3. EVALUATE $\hat{P}$: Evaluate the population $\hat{P}$ using the surrogate model.
        Step 4. FIND AN APPROXIMATION TO $PF$: By using MOEA/D, the surrogate model and the population $\hat{P}$, obtain the approximation $\hat{P}$ to $PF$.
        Step 5. SELECT POINTS FOR UPDATING $T_{set}$: See section III-B.
        Step 6. UPDATE POPULATION $\hat{P}$: See section III-C.
        Step 7. LOCAL SEARCH: Apply nonlinear simplex search by using the training set $T_{set}$, see section IV-B and Algorithm 3.
    end
    return $A$;
end
```

IV. THE PROPOSED MOEA/D-RBF WITH A LOCAL SEARCH MECHANISM

A. General Framework

Our proposed MOEA/D-RBF with Local Search (MOEA/D-RBF+LS) decomposes the problem (1) into N single-objective optimization problems. MOEA/D-RBF uses a well-distributed set of N weight vectors $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ to define a set of single-objective optimization subproblems by using the PBI approach. Each subproblem is solved by MOEA/D, which is assisted by RBF networks. After each iteration of MOEA/D, the local search procedure is applied. Algorithm 2 shows the general framework of our proposed MOEA/D-RBF+LS. In the following sections, we describe in detail the proposed local search mechanism adopted by our approach.

B. Local Search Mechanism

As indicated before, the local search adopted by MOEA/D-RBF+LS is based on the Nelder and Mead algorithm [15] (also known as Nonlinear Simplex Search (NSS)). The search done by the NSS is based on geometric operations on a set of points, which define an n-dimensional polygon called "simplex", where n is the number of decision variables of the problem. The effectiveness of NSS as a local search mechanism in MOEAs has been shown by several authors, see for example [11], [22], [26]. Here, we take the advantage of the decomposition approach to direct the search of the NSS. In this way, the NSS is employed for minimizing a subproblem defined by a weight vector using the PBI approach. In the following, we present in detail the components of our local search engine outlined in Algorithms 2 and 3.

Algorithm 3: Use of Local Search

```
Input:
a stopping criterion;
$T_{set}$: The training set used by MOEA/D-RBF+LS.
$W_{ls} = \{\mathbf{w}_i, \ldots, \mathbf{w}_N\}$: A well-distributed set of weight vectors for the local search.
$S_t$: the similarity threshold for the local search;
$E_{ls}$: the maximum number of evaluations for the local search.
Output:
$T_{set}$: the updated training set $T_{set}$.
begin
    Step 1. DEFINING THE POPULATION $P_{ls}$: Using the weight set $W_{ls}$ and the training set $T_{set}$, define the population $P_{ls}$ from which the local search is performed, see Section IV-B1;
    Step 2 DEFINING THE SEARCH DIRECTION AND THE INITIAL SOLUTION: Define the search direction and the initial solutions from which the local search starts, according to Section IV-B2
    Step 3. CHECKING SIMILARITY: Obtain the similarity ($S_{ls}$) between $\mathbf{p}_{ini}$ and the previous initial solution ($\mathbf{p}'_{ini}$) for the local search, see Section IV-B3;
    if there are enough resources and $S_t < S_{ls}$ then
        Step 4. BUILDING THE SIMPLEX: Build the initial simplex for the nonlinear simplex search, see Section IV-B4;
        Step 5. DEFORMING THE SIMPLEX: Perform any movement (reflection, contraction or expansion) for obtaining $\mathbf{p}_{new}$ according to Nelder and Mead's method, see Section IV-B5;
        Step 6. UPDATING THE POPULATION AND THE EXTERNAL ARCHIVE: Update the population $P_{ls}$ and the external archive $A$ using the new solution $\mathbf{p}_{new}$ according to the rules presented in Section IV-B6.
        Step 7. STOPPING CRITERION: If the stopping criterion is satisfied then stop the local search and go to Step 7. Otherwise, go to Step 4, see Section IV-B7.
    end
    Step 8. UPDATING THE TRAINING SET: Update the training set $T_{set}$ according to the updating scheme, see Section IV-B8.
end
```

1) Defining the Population P_{ls}: At the beginning, a new population P_{ls} of N_{ls} individuals is defined in order to direct the local search. Let W_{ls} and T_{set} be a well-distributed set of weight vectors and the training set, respectively. P_{ls} is stated by choosing different solutions $\mathbf{x}^t \in T_{set}$ such that they minimize:

$$g(\mathbf{x^t}|\mathbf{w_i}, \mathbf{z}^\star), \text{for each } \mathbf{w}_i^{ls} \in W_{ls}$$

The cardinality of W_{set} should be much less that the cardinality of the weight set W (which directs the search of MOEA/D-RBF+LS), i.e., $|W_{ls}| << |W|$. With that, a small portion of search directions are considered by the local search engine, in order to obtain well-spread solutions along the Pareto Front.

2) Defining the Search Direction and the Initial Solution: The proposed local search mechanism, approximates solutions to the maximun bulge (sometimes called knee) of the Pareto front. Therefore, the local search is focused on minimizing the subproblem that approximates the solutions lying on the knee of the Pareto front. Thus, the search direction is defined by the weighting vector:

$$\mathbf{w}_s = (1/k, \ldots, 1/k)^T$$

where k is the number of objective functions. Considering the use of the PBI approach, the penalty value θ is set to $\theta = 10$.

Let A be the set of nondominated solutions found during the search of MOEA/D-RBF+LS. Let $\mathbf{w}_s$ be the weighting vector that defines the search direction for the nonlinear simplex search. The solution $\mathbf{p}_{ini}$ which starts the search is defined by:

$$\mathbf{p}_{ini} = \mathbf{x} \in A, \text{ such that minimizes: } g(\mathbf{x}|\mathbf{w}_s, \mathbf{z}^{\star})$$

Solution $\mathbf{p}_{ini}$ represents not only the initial search point, but also the simplex head from which the simplex will be built.

3) Checking Similarity: The NSS explores the neighborhood of the solution $\mathbf{p}_{ini} \in A$. Since the simplex search is applied after each iteration of the MOEA/D, most of the time, the initial solution $\mathbf{p}_{ini}$ does not change its position from one generation to another. For this reason, the proposed local search mechanism stores a record ($\mathbf{p}'_{ini}$) of the last position from which the nonlinear simplex search starts. At the beginning of the execution of MOEA/D-RBF+LS, the initial position record is set as empty, that is: $\mathbf{p}'_{ini} = \emptyset$. Once the simplex search is performed, the initial solution is stored in the historical record, i.e., $\mathbf{p}'_{ini} = \mathbf{p}_{ini}$. In this way, for the next call of the local search, a previous comparison of similarity is performed. That is, the local search will be applied, if and only if, $||\mathbf{p}_{ini} - \mathbf{p}'_{ini}|| > S_{ls}$, where S_{ls} represents the similarity threshold. Both the updating of the historical record and the similarity operator are performed for each initial solution $\mathbf{p}_{ini}$ which minimizes the subproblem defined by $\mathbf{w}_s$. As in [23], we adopted a similarity threshold $S_{ls} = 0.001$.

4) Building the Simplex: Let A be the set of nondominated solutions found during the search of MOEA/D-RBF+LS. Then, the simplex $\mathbf{\Delta}$ is built in three different ways, depending of the cardinality of A.

i. $|A| = 1$: Set $\sigma = (0.01, \ldots, 0.01)^T$ and the simplex is defined as:

$$\mathbf{\Delta} = \{\mathbf{a}, \Delta_2, \ldots, \Delta_{n+1}\}$$

where $\mathbf{a} \in A \subset \Omega$ and the remaining n vertices $\Delta_i \in \Omega$ $(i = 1, \ldots, n)$ are generated by using a low-discrepancy sequence. In our study, we adopted the Hammersley sequence [7] to generate a well-distributed sampling of solutions in a determined search space. The search space is defined by:

$$\begin{aligned} \mathbf{L}_{bound} &= \mathbf{a} - \sigma \\ \mathbf{U}_{bound} &= \mathbf{a} + \sigma \end{aligned}$$

In this way, the vertices are generated by means of the Hammersley sequence using as bounds $\mathbf{L}_{bound}$ and $\mathbf{U}_{bound}$.

ii. $1 < |A| < (n+1)$: The simplex is defined by using all solutions in A and the remaining $l = (n+1) - |A|$ solutions are generated by using the Hammersley sequence. However, the bounds are defined as:

$$\begin{aligned} \mathbf{L}_{bound} &= \mathbf{m} - \sigma \\ \mathbf{U}_{bound} &= \mathbf{m} + \sigma \end{aligned}$$

where $(\mathbf{m})$ and (σ) are the average and the standard deviation of the solutions contained in A, respectively. $\mathbf{L}_{bound}$ and $\mathbf{U}_{bound}$ are the lower and upper bounds of the new search space, respectively.

iii. $|A| \geq (n+1)$: In this case, the simplex is built by choosing in a random way, $n+1$ solutions taken from A.

5) Deforming the Simplex: Let $\mathbf{w}_s$ be the weight vector that defines the search direction for the NSS. Let $\mathbf{\Delta}$ be the simplex defined by the above description. The simplex search will be focused on minimizing the subproblem defined by the weighting vector $\mathbf{w}_s$. At each iteration of the simplex search, the $n+1$ vertices of the simplex $\mathbf{\Delta}$ are sorted according to their value for the subproblem that it tries to minimize (the best value is the first element). In this way, a movement into the simplex is performed for generating the new solution $\mathbf{p}_{new}$. The movements are calculated according to the equations provided by Nelder and Mead [15]. Each movement is controlled by three scalar parameters: **reflection** (ρ), **expansion** (χ) and **contraction** (γ).

The NSS was conceived for unbounded problems. When dealing with bounded variables, the created solutions can be located outside the allowable bounds after some movements of the NSS. In order to deal with this, we bias the new solution if any component of $\mathbf{p}_{new}$ lies outside the bounds according to:

$$\mathbf{p}_{new}^{(j)} = \begin{cases} \mathbf{L}_{bound}^{(j)} & \text{, if } \mathbf{p}_{new}^{(j)} < \mathbf{L}_{bound}^{(j)} \\ \mathbf{U}_{bound}^{(j)} & \text{, if } \mathbf{p}_{new}^{(j)} > \mathbf{U}_{bound}^{(j)} \\ \mathbf{p}_{new}^{(j)} & \text{, otherwise.} \end{cases} \quad (4)$$

where $\mathbf{L}_{bound}^{(j)}$ and $\mathbf{U}_{bound}^{(j)}$ are the lower and upper bounds of the j^{th} parameter of $\mathbf{p}_{new}$, respectively.

6) Updating the Population and the External Archive: The information provided by the local search mechanism is introduced into the population P_{ls}. Since we are dealing with MOPs, the new solution generated by any movement of the simplex search could be better than more than one solution in the current population. Thus, we adopt the following mechanism in which more than one solution from the population could be replaced.

Let $\mathbf{p}_{new}$ be the solution generated by any movement of the NSS. Let $B(\mathbf{w}_s)$ and W_{ls} be the neighborhood of the T closest weighting vectors to $\mathbf{w}_s$, and the well-distributed set of all weighting vectors, respectively. We define:

$$Q = \begin{cases} B(\mathbf{w}_s) & \text{, if } r < \delta \\ W & \text{otherwise} \end{cases}$$

where r is a random number having a uniform distribution. In this work, we use $\delta = 0.5$.

The population P_{ls} is updated by replacing at most R_{ls} solutions from P_{ls} such that, $g(\mathbf{p}_{new}|\mathbf{w}_i, z) < g(\mathbf{x}_i|\mathbf{w}_i, z)$, where $\mathbf{w}_i \in Q$ and $\mathbf{x}_i \in P_{ls}$, such that $\mathbf{x}_i$ minimizes the subproblem defined by $\mathbf{w}_i$. In our study, we set $R_{ls} = 15$ as the maximum number of solutions to replace.

The external archive A contains the nodominated solutions found during the search of MOEA/D-RBF+LS. For each new solution $\mathbf{p}_{new}$, the external archive is updated by removing from A all the solutions dominated by $\mathbf{p}_{new}$, and then, $\mathbf{p}_{new}$ is stored in A if no solutions in A dominate $\mathbf{p}_{new}$.

7) Stopping Criterion: The local search procedure is limited to a maximum number of fitness function evaluations defined by E_{ls}. In this way, the proposed local search has the following stopping criteria:

1) If the nonlinear simplex search overcomes the maximum number of evaluations (E_{ls}) or there are not enough resources for continuing the search of MOEA/D+LS, the local search is stopped.
2) The search could be inefficient if the simplex has been deformed so that it has collapsed into a region in which there are no local minima. According to Lagarias et al. [12] the simplex search finds a better solution in at most $n+1$ iterations (at least in convex functions with low dimensionality). Therefore, if the simplex search does not find a better value for the subproblem defined by $\mathbf{w}_s$ in $n+1$ iterations, we stop the search. Otherwise, we perform another movement into the simplex by going to **Step 3** of Algorithm 3.

8) Updating the Training Set: The knowledge obtained by the local search is introduced to MOEA/D-RBF+LS by updating the training set T_{set}. The maximum number of solutions in the training set T_{set} is defined by the parameter N_t. The updating of T_{set} is carried out by defining a well-distributed set of N_t weight vectors $W_t = \{\mathbf{w}_1^t, \ldots, \mathbf{w}_{N_t}^t\}$. Therefore, the best N_t different solutions from $T = \{T_{set} \cup A\}$, such that they minimize the subproblems defined by each weight vector $\mathbf{w}_i^t \in W_t$, are used to update T_{set}. With this, the nondominated solutions found by the local search are included in T_{set} and the model can be improved even if it has been previously misinformed.

V. TEST PROBLEMS

A. Standard Test Problems

In order to assess the performance of our proposed MOEA/D-RBF+LS, we compare its results with respect to those obtained by MOEA/D-RBF [24]. We adopted five test problems whose Pareto fronts have different characteristics including convexity, concavity, disconnections and multi-modality. The bi-objective test suite of Zitzler-Deb-Thiele (ZDT) [27] (except for ZDT5, which is a binary problem) is adopted. We used 30 decision variables for problems from ZDT1 to ZTD3, while ZDT4 and ZDT6 were tested using 10 decision variables, as suggested in [27].

B. Airfoil Shape Optimization: A case study

The case study presented here, consists of the multi-objective optimization of an airfoil shape problem adapted from [19] (denoted as MOPRW). This problem corresponds to the airfoil shape optimization of a standard-class glider, aiming to obtain an optimum performance for a sailplane. In this study, the trade-off among two aerodynamic objectives is evaluated using our proposed approach, and its results are compared with respect to those obtained by MOEA/D-RBF.

1) Problem Statement: Two conflicting objective functions are defined in terms of a sailplane average weight and operating conditions [19]. They are defined as:

TABLE I. PARAMETER RANGES FOR MODIFIED PARSEC AIRFOIL REPRESENTATION

Design Variable	Lower Bound	Upper Bound
r_{leup}	0.0085	0.0126
r_{lelo}	0.0020	0.0040
α_{te}	7.0000	10.0000
β_{te}	10.0000	14.0000
Z_{te}	-0.0060	-0.0030
ΔZ_{te}	0.0025	0.0050
X_{up}	0.4100	0.4600
Z_{up}	0.1100	0.1300
Z_{xxup}	-0.9000	-0.7000
X_{lo}	0.2000	0.2600
Z_{lo}	-0.0230	-0.0150
Z_{xxlo}	0.0500	0.2000

i) Minimize: C_D/C_L
s.t. $C_L = 0.63$, $Re = 2.04 \cdot 10^6$, $M = 0.12$

ii) Minimize: $C_D/C_L^{3/2}$
s.t. $C_L = 1.05$, $Re = 1.29 \cdot 10^6$, $M = 0.08$

where C_D/C_L and $C_D/C_L^{3/2}$ correspond to the inverse of the glider's gliding ratio and sink rate, respectively. Both are important performance measures for this aerodynamic optimization problem. C_D and C_L are the drag and lift coefficients, respectively.

The aim is to maximize the gliding ratio (C_L/C_D) for objective (*i*), while minimizing the sink rate in objective (*ii*). Each of these objectives is evaluated at different prescribed flight conditions, given in terms of Mach and Reynolds numbers. The aim of solving this MOP is to find a better airfoil shape, which improves a reference design.

2) Geometry Parameterization: In the present case study, the PARSEC airfoil representation [18] was adopted. Fig. 1 illustrates the 11 basic parameters used for this representation: r_{le} leading edge radius, X_{up}/X_{lo} location of maximum thickness for upper/lower surfaces, Z_{up}/Z_{lo} maximum thickness for upper/lower surfaces, Z_{xxup}/Z_{xxlo} curvature for upper/lower surfaces, at maximum thickness locations, Z_{te} trailing edge coordinate, ΔZ_{te} trailing edge thickness, α_{te} trailing edge direction, and β_{te} trailing edge wedge angle.

For the present case study, the modified PARSEC geometry representation adopted allows us to define independently the leading edge radius, both for upper and lower surfaces. Thus, a total of 12 variables are used. Their allowable ranges are defined in Table I. The PARSEC airfoil geometry representation uses a linear combination of shape functions for defining the upper and lower surfaces. These linear combinations are given by:

$$Z_{upper} = \sum_{n=1}^{6} a_n x^{\frac{n-1}{2}}, \quad Z_{lower} = \sum_{n=1}^{6} b_n x^{\frac{n-1}{2}} \qquad (5)$$

In the above equations, the coefficients a_n, and b_n are determined as functions of the 12 described geometric parameters, by solving two systems of linear equations, one for each surface. It is important to note that the geometric parameters r_{leup}/r_{lelo}, X_{up}/X_{lo}, Z_{up}/Z_{lo}, Z_{xxup}/Z_{xxlo}, Z_{te}, ΔZ_{te}, α_{te}, and β_{te} are the actual design variables in the optimization process, and that the coefficients a_n, b_n serve as intermediate variables for interpolating the airfoil's coordinates, which are used by the CFD solver (we used the Xfoil CFD code [3]) for its discretization process.

VI. COMPARISON OF RESULTS

A. Performance Assessment

To assess the performance of our proposed MOEA/D-RBF+LS and MOEA/D-RBF on the test problems adopted, the **Hypervolume** (I_H) indicator was employed [28]. This performance measure is Pareto compliant [29], and quantifies both approximation and maximum spread of nondominated solutions along the Pareto front. A high I_H value, indicates that the approximation P is close to PF and has a good spread towards the extreme portions of the Pareto front. The interested reader is referred to [28] for a more detailed description of this performance measure.

B. Experimental Setup

As indicated before, the proposed approach is compared with respect to the original MOEA/D-RBF. For each MOP, 30 independent runs were performed with each algorithm. Each algorithm was restricted to 1,000 fitness function evaluations. For the airfoil design problem, the search was restricted to 5,000 fitness function evaluations.

The parameters used for MOEA/D, which is employed by MOEA/D-RBF and MOEA/D-RBF+LS, were set as in [24]. This is because there is empirical evidence that indicates that these are the most appropriate parameters for solving the ZDT test suite, see [24]. The weight vectors for the algorithms were generated as in [24], i.e., the settings of N and $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ is controlled by a parameter H. More precisely, $\mathbf{w}_1, \ldots, \mathbf{w}_N$ are all the weight vectors in which each individual weight takes a value from:

$$\left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}$$

Therefore, the number of such vectors in W is given by $N = C^{k-1}_{H+k-1}$, where k is the number of objective functions (for the test problems adopted $k = 2$). For MOEA/D-RBF and MOEA/D-RBF+LS, the set W was defined with $H = 299$, i.e., 300 weight vectors. The set W_t was generated with $H = 10n - 1$. Therefore, $N_t = 10n$ weight vectors (which define the cardinality of the training set), where n is the number of decision variables of the MOP. The set W_s uses $H = 9$, i.e., $N_s = 10$ weight vectors. Note that these values of parameters are the ones used by MOEA/D-RBF in [23].

For the local search, the set W_{ls} was generated using $H = 99$, therefore $N_{ls} = 100$. The NSS was performed using $\rho = 1, \chi = 2$ and $\gamma = 1/2$, for the reflection, expansion and contraction, respectively. The maximum number of solutions to be replaced was set to $R_{ls} = 15$ and the maximum number of fitness function evaluations was set to $E_{ls} = 2(n + 1)$. Finally, the similarity threshold was set to $S_{ls} = 0.01$. The execution of the algorithms was carried out on a computer with a 2.66GHz processor and 4GB in RAM.

As indicated before, the algorithms were evaluated using the I_H performance measure. The results obtained are summarized in Table II. This table display both the $average$ and the standard deviation (σ) of the I_H indicator for each MOP, respectively. The reference vector $\mathbf{r}$ used for computing I_H, for each MOP, is shown in Table II. For an easier interpretation, the best results are presented in **boldface** for each test problem adopted.

C. Discussion of Results

1) ZDT Test Problems: Table II shows the results obtained for the I_H indicator when the algorithms were tested on the ZDT test problems. From this table it is possible to see that MOEA/D-RBF+LS obtained a better approximation to PF than the one achieved by MOEA/D-RBF in most of the test problems adopted. The exception was ZDT1 where MOEA/D-RBF was better than MOEA/D-RBF+LS. However, MOEA/D-RBF was not significantly better than MOEA/D-RBF+LS. The performance of MOEA/D-RBF+LS and MOEA/D-RBF was very similar for ZDT1 and ZDT2. The differences were more significant for ZDT3, ZDT4 and ZDT6. These last problems have special features that deteriorate the good performance of surrogate models. ZDT3 is a problem whose PF consists of several noncontiguous convex parts. ZDT4 is multi-modal problem, which causes difficulties to model the search space in a suitable way. ZDT6 has two difficulties caused by the nonuniformity of the search space: first, the Pareto optimal solutions are nonuniformly distributed along the PF; second, the density of the solutions is lowest near the PF and gets higher as we move away from the PF. These features evidently present a major obstacle to the surrogate model employed by MOEA/D-RBF. However, the use of local search for these problems, improved the performance of MOEA/RBF. In fact, MOEA/D-RBF+LS obtained better approximations to the PF for these MOPs, and in some cases, such as in ZDT4 and ZDT6, it was significantly better.

2) Airfoil Design Problem: For this particular problem, the features of the PF are unknown. According to the results presented in Table II, we can see that MOEA/D-RBF+LS obtained better I_H values than those reached by MOEA/D-RBF. This means that our proposed MOEA/D-RBF+LS obtained a better approximation and spread of solutions along the PF than MOEA/D-RBF.

According to the results reported in [23], the original MOEA/D employed, on average, 5,050 seconds to achieve convergence with 5,000 fitness function evaluations. MOEA/D-RBF and MOEA/D-RBF+LS employed, on average, between 1,900 and 2,000 seconds to achieve a value in the I_H indicator similar to the one reported by MOEA/D, respectively. Therefore, we argue that our proposed MOEA/D-RBF is a good choice for dealing with computationally expensive MOPs.

Fig. 1. PARSEC airfoil parameterization.

TABLE II. RESULTS OF THE I_H METRIC FOR MOEA/D-RBF+LS AND MOEA/D-RBF

MOP	MOEA/D-RBF+LS	MOEA/D-RBF	Reference vector **r**
	$average$ (σ)	$average$ (σ)	
ZDT1	0.868197 (0.002837)	**0.870908** (0.000371)	$(1.1, 1.1)^T$
ZDT2	**0.536389** (0.004921)	0.536265 (0.000593)	$(1.1, 1.1)^T$
ZDT3	**0.876380** (0.102611)	0.837894 (0.179280)	$(1.1, 1.1)^T$
ZDT4	**12.441923** (30.715277)	5.739229 (30.906695)	$(30.0, 30.0)^T$
ZDT6	**96.299610** (0.761493)	95.313012 (1.271933)	$(10, 10)^T$
MOPRW	**2.6818676e-07** (6.417924e-09)	2.493786e-07 (6.483342e-09)	$(0.007610, 0.005236)^T$

VII. CONCLUSIONS AND FUTURE WORK

The effectiveness of MOEA/D-RBF was tested in [23], where it was compared with respect to the original MOEA/D and a current state-of-the-art MOEA assisted by surrogate models (the MOEA/D-EGO [25]). Here, we have introduced an extension of MOEA/D-RBF which includes a local search mechanism in order to improve the convergence to the Pareto front, when a low number of fitness function evaluations is used. The proposed MOEA/D-RBF+LS was able to improve the convergence of MOEA/D-RBF, when the search was limited to a low number of fitness function evaluations. We also validated our proposed approach with a real-world computationally expensive MOP: an airfoil design problem. The obtained results have shown that MOEA/D-RBF+LS is a viable choice to deal with MOPs having different features, and the applicability to real-world applications could speed up convergence to the PF in comparison to conventional MOEAs.

As part of our future work, we plan to use our approach in problems having three or more objectives, which represent a challenge to MOEAs assisted by surrogate models. Also, we intend to explore the use of other mathematical programming techniques in the local search mechanism. Finally, we are also interested in testing our approach with more real-world problems having a higher number of decision variables, and this is, indeed, part of our ongoing research.

REFERENCES

[1] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2nd edition, 2007.

[2] I. Das and J. E. Dennis. Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[3] M. Drela. XFOIL: An Analysis and Design System for Low Reynolds Number Aerodynamics. In *Conference on Low Reynolds Number Aerodynamics*, University Of Notre Dame, IN, June 1989.

[4] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition edition, June 2005.

[5] M. T. M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

[6] C. A. Georgopoulou and K. C. Giannakoglou. A multi-objective metamodel-assisted memetic algorithm with strengthbased local refinement. *Engineering Optimization*, 41(10):909–923, 2009.

[7] J. M. Hammersley. Monte-Carlo methods for solving multivariable problems. *Annals of the New York Academy of Science*, 86:844–874, 1960.

[8] R. Hooke and T. A. Jeeves. “direct search” solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, 1961.

[9] A. Isaacs, T. Ray, and W. Smith. An evolutionary algorithm with spatially distributed surrogates for multiobjective optimization. In *ACAL*, pages 257–268, 2007.

[10] J. Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, January 2006.

[11] P. Koduru, S. Das, and S. M. Welch. Multi-Objective Hybrid PSO Using ϵ-Fuzzy Dominance. In D. Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO’2007)*, volume 1, pages 853–860, London, UK, July 2007. ACM Press.

[12] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9:112–147, 1998.

[13] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[14] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachuisetts, 1999.

[15] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7:308–313, 1965.

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

[17] W. Peng and Q. Zhang. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *IEEE International Conference on Granular Computing, 2008. GrC 2008*, pages 534 –537, 2008.

[18] H. Sobieczky. Parametric Airfoils and Wings. In K. Fuji and G. S. Dulikravich, editors, *Notes on Numerical Fluid Mechanics, Vol.. 68*, pages 71–88, Wiesbaden, 1998. Vieweg Verlag.

[19] A. Szöllös, M. Smíd, and J. Hájek. Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. *Advances in Engineering Software*, 40(6):419–430, 2009.

[20] S. Zapotecas Martínez and C. A. Coello Coello. A Memetic Algorithm with Non Gradient-Based Local Search Assisted by a Meta-Model. In *PPSN XI*, volume 6238, pages 576–585. Springer. Lecture Notes in Computer Science, September 2010.

[21] S. Zapotecas Martínez and C. A. Coello Coello. A Multi-objective Particle Swarm Optimizer Based on Decomposition. In *GECCO’2011*, pages 69–76, Dublin, Ireland, July 2011. ACM.

[22] S. Zapotecas Martínez and C. A. Coello Coello. A Direct Local Search Mechanism for Decomposition-based Multi-Objective Evolutionary Algorithms. In *CEC’2012*, pages 3431–3438, Brisbane, Australia, June 2012. IEEE Press.

[23] S. Zapotecas Martínez and C. A. Coello Coello. MOEA/D assisted by RBF Networks for Expensive Multi-Objective Optimization Problems. Technical Report EVOCINV-02-2013, Evolutionary Computation Group at CINVESTAV, Departamento de Computación, CINVESTAV-IPN, México, February 2013.

[24] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[25] Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model. *Evolutionary Computation, IEEE Transactions on*, 14(3):456 –474, june 2010.

[26] X. Zhong, W. Fan, J. Lin, and Z. Zhao. Hybrid non-dominated sorting differential evolutionary algorithm with nelder-mead. In *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*, volume 1, pages 306 –311, December 2010.

[27] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evo-

lutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[28] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[29] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.