Jyväskylän yliopiston julkaisuarkisto
Jyväskylä University Digital Archive

UNIVERSITY OF JYVÄSKYLÄ

This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.

Author(s): Sindhya, Karthik; Ruuska, Sauli; Haanpää, Tomi; Miettinen, Kaisa

Title: A New Hybrid Mutation Operator for Multiobjective Optimization with Differential Evolution

Year: 2011

Version:

Please cite the original version:

Sindhya, K., Ruuska, S., Haanpää, T., & Miettinen, K. (2011). A New Hybrid Mutation Operator for Multiobjective Optimization with Differential Evolution. Soft Computing, 15(10), 2041-2055. https://doi.org/10.1007/s00500-011-0704-5

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Noname manuscript No.
(will be inserted by the editor)

A New Hybrid Mutation Operator for Multiobjective Optimization with Differential Evolution

Karthik Sindhya · Sauli Ruuska · Tomi Haanpää · Kaisa Miettinen

Received: date / Accepted: date

Abstract Differential evolution has become one of the most widely used evolutionary algorithms in multiobjective optimization. Its linear mutation operator is a simple and powerful mechanism to generate trial vectors. However, the performance of the mutation operator can be improved by including a nonlinear part. In this paper, we propose a new hybrid mutation operator consisting of a polynomial based operator with nonlinear curve tracking capabilities and the differential evolution's

Karthik Sindhya
Department of Mathematical Information Technology, P.O. Box 35 (Agora),
FI-40014 University of Jyväskylä, Finland
E-mail: karthik.sindhya@jyu.fi

S. Ruuska
Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
E-mail: sauli.ruuska@jyu.fi

T. Haanpää
Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
E-mail: tomi.v.haanpaa@jyu.fi

K. Miettinen
Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
E-mail: kaisa.miettinen@jyu.fi

original mutation operator, to be efficiently able to handle various interdependencies between decision variables. The resulting hybrid operator is straightforward to implement and can be used within most evolutionary algorithms. Particularly, it can be used as a replacement in all algorithms utilizing the original mutation operator of differential evolution. We demonstrate how the new hybrid operator can be used by incorporating it into MOEA/D, a winning evolutionary multiobjective algorithm in a recent competition. The usefulness of the hybrid operator is demonstrated with extensive numerical experiments showing improvements in performance compared to the previous state of the art.

Keywords Evolutionary Algorithms · DE · Nonlinear · Multi-criteria Optimization · Polynomial · Pareto Optimality · MOEA/D.

1 Introduction

In evolutionary multiobjective optimization (EMO), nature-inspired numerical methods known as evolutionary algorithms (EAs) are applied to solve optimization problems with multiple, conflicting objectives. Unlike a single-objective optimization problem, a multiobjective optimization problem (MOP) does not, in general, have a unique optimal solution. Instead, the optimal solutions to a MOP constitute a possibly infinite set of compromise solutions, known as Pareto optimal solutions, which can be ordered only by subjective preferences. Intuitively, the aim in EMO is to represent the range of tradeoffs within the Pareto optimal solutions using a fixed-size population of nondominated solutions.

The two main concerns in designing an EMO method are the optimality of obtained solutions (convergence), and the representativeness of the solutions (diversity). Thus, an EMO method must maintain diversity in the population not only to prevent premature convergence, but because the final population should represent a wide range of different nondominated solutions. Ensuring diversity in the objective space is a nontrivial task and perhaps for this reason there has been an emphasis on research towards selection operators—which operate in the objective space—instead of the mutation and crossover operators. In many cases, the mu-

tation and crossover operators used in an EMO method have been developed for single-objective optimization (see, for example, Coello et al (2007); Deb (2001)). This fact has already been pointed out in Jin and Sendhoff (2003); Zhang et al (2008). In Zhang et al (2008), it is used, together with an assumed regularity of the solution set in the decision space, to motivate a multiobjective optimization method modeling explicitly the interdependencies between decision variables.

In single-objective optimization, differential evolution (DE) has become a widely used method (Price et al (2005); Storn and Price (1996)). An appealing feature of DE is its extremely uncomplicated, self-adapting mutation operator which is based on random perturbations obtained by sampling difference vectors from the population of decision vectors maintained by the algorithm. The mutation operator of DE is capable of implicitly detecting and exploiting linear interdependencies between decision variables, but does not cope as well with more complicated, nonlinear interdependencies (Ruuska and Aittokoski (2008)). We say that a MOP involves nonlinear interdependencies between decision variables if the Pareto-optimal solutions in the decision space do not fall on a plane of lower dimension than the space itself.

Several variants of DE have been proposed in the literature also for multiobjective optimization including Abbass (2002); Babu and Jehan (2003); Kukkonen and Lampinen (2005); Robic and Filipic (2005); Santana-Quintero and Coello (2005). For a review, see, for example, Price et al (2005); Santana-Quintero and Coello (2005). The existing multiobjective variants of DE have also proven successful in solving many of the multiobjective test problems available (Kukkonen and Lampinen (2006); Zhang et al (2009a)). However, as pointed out in Jin and Sendhoff (2003); Okabe et al (2004), in many widely used multiobjective test problems, the solution set in the decision space can be defined by piecewise linear functions even though the objective functions as such are nonlinear. This raises the question whether part of the success of the DE-based EMO methods could be due to the capability of their mutation operator to exploit linear interdependencies between decision variables in a problem, and whether the efficiency of these methods in solving problems with nonlinear interdependencies could be improved by changes in the mutation operator only.

Different mutation operators have been proposed in the literature. Among others, a trigonometric mutation operator was proposed by Fan and Lampinen (2003). Here, a hypertriangle formed by three vectors is used to bias the perturbation of a target vector towards the vector providing the lowest function value. In Cauchy mutation (Ali and Pant (2010)), a decision vector is perturbed by using a Cauchy distributed random variable, with the hope of pulling them from a local basis of attraction. Liu and Lampinen (2005) proposed a fuzzy adaptive differential evolution algorithm, which uses fuzzy logic controllers to adapt search parameters (scaling factor, crossover ratio and population size). In PDE-PEDA proposed by Wang et al (2009) an estimation of distribution algorithm (global operator) and a linear DE crossover (local operator) is used with a self-adaptive probability. The self-adaptive probability is used here to balance the global and local information. Lara et al (2010) proposed a local search strategy (HCS) using the geometry of the directional cones to generate solutions both towards and along the Pareto set. Finally, let us mention a hybrid DE operator proposed by Kaelo and Ali (2007). This mutation operator using both the linear mutation operator of DE and an electromagnetism-like algorithm (using an attraction-repulsion technique to move closer to global minima) is used with a pre-fixed probability. Overall, one can say that not much attention has been paid in the DE literature on handling both linear and nonlinear interdependencies between decision variables.

In this paper, our motivation is to propose a new hybrid mutation operator which can robustly handle both linear and nonlinear interdependencies between decision variables, simultaneously retaining the simplicity of the linear differential evolution mutation operator. In practice, no a priori knowledge exists about the interdependencies between decision variables. Hence, the proposed operator is a combination of the DE's linear mutation operator and a new polynomial part to handle nonlinear interdependencies between decision variables, which can be used as a drop-in replacement for DE's mutation operator.

The so-called curvature detection in the polynomial part is based on polynomial approximation which is used to guide the generation of new trial vectors. One can say that there is nothing new in using polynomials in optimization; especially

quadratic polynomials are frequently used for line search and other purposes (see, e.g., Bazaraa et al (2006); Nocedal and Wright (1999)). Polynomials have also been used for trial vector generation in population-based algorithms before (see, e.g., Ali et al (1997); Schütze et al (2007)). Our use of polynomials, however, differs from the current practice in that we do not use polynomials to model objective functions, but the interdependencies between decision variables of the problem. In particular, new trial points are not determined by the extreme points of the approximations as is common in line search procedures.

The rest of the paper is structured as follows. In Section 2, we discuss the basics of multiobjective optimization and DE as well as introduce the notation used in this paper. Then, in Section III we propose a new polynomial part and, finally, a new hybrid operator which utilizes both the original operator of DE as well as the new polynomial part. In Section IV we demonstrate the potential and usefulness of the new hybrid operator when solving multiobjective optimization problems. With a versatile set of test problems we show how the performance of a linear mutation based MOEA/D algorithm can be improved with our hybrid operator. This should encourage further research in this direction. Finally, the paper is concluded in Section 5.

2 Notation and Background

In this section, we present the notation and background necessary for the rest of the paper. First we formulate a MOP with the relevant definitions. Then we provide a brief review of the functioning and properties of the DE.

2.1 Basics of Multiobjective Optimization

We consider a MOP

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})) \\ &\text{subject to } \mathbf{x} \in S \end{aligned} \tag{1}$$

with a feasible set $S \subset \mathbb{R}^n$ and $k \geq 2$ conflicting objective functions $f_i \colon S \to \mathbb{R}$. The n-dimensional vectors $\mathbf{x} \in S$ are called *decision vectors* and their images $\mathbf{z} = \mathbf{f}(\mathbf{x})$ *objective vectors*. The set of attainable objective vectors is denoted by Z, $Z = \mathbf{f}(S)$.

An objective vector $\mathbf{z} \in Z$ is called *Pareto optimal* if there does not exist a vector $\bar{\mathbf{z}} \in Z$ such that $\bar{z}_i \leq z_i$ $(i = 1, 2, \ldots, k)$ with at least one strict inequality. Naturally, a decision vector $\mathbf{x} \in S$ is Pareto optimal if its image $\mathbf{f}(\mathbf{x})$ is Pareto optimal. In what follows, the set of Pareto-optimal decision vectors is called the *Pareto set*, and the set of Pareto-optimal objective vectors the *Pareto front*. For two decision vectors $\mathbf{x}^1, \mathbf{x}^2 \in S$, $\mathbf{x}^1$ is said to *dominate* $\mathbf{x}^2$ if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ $\forall\, i = 1, \ldots, n$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one index j. In the context of EMO methods, the subset of vectors in a population not dominated by any other vector in the population is called the *nondominated set*. Typically, EMO algorithms aim at generating nondominated solutions representing the Pareto front as well as possible (i.e., both being close to the Pareto front and reflecting different tradeoffs).

2.2 Basics of Differential Evolution

DE is a stochastic, population-based direct search method introduced by Storn and Price for optimizing a real-valued function of continuous variables (Price et al (2005); Storn and Price (1996)). It is distinguished from other EAs mostly by its remarkably simple trial vector generation scheme in which a scaled difference of vectors originating from a fixed-size population of decision vectors is used. In spite of its simplicity, DE has demonstrated its capability in finding good solutions to a wide variety of different benchmark problems as well as practical optimization problems while comparing favorably to other EAs in terms of convergence speed (Price et al (2005)).

The basic step of DE consists of updating a population of $p + 1$ decision vectors $P = \{\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^p\}$. For each vector $\mathbf{x}^t \in P$ in turn, a *trial vector* $\hat{\mathbf{x}}^t$ is generated. The vector $\mathbf{x}^t$ is called a *target vector* because it will be replaced in P by the trial vector $\hat{\mathbf{x}}^t$ for the next iteration if $\hat{\mathbf{x}}^t$ yields a better objective function value than $\mathbf{x}^t$. First, a subset of three decision vectors $Q = \{\mathbf{x}^{r_1}, \mathbf{x}^{r_2}, \mathbf{x}^{r_3}\} \subset P$ is selected using three

distinct random indices r_i, $0 \leq r_i \leq p$, $i = 1, 2, 3$. In the original DE mutation, a trial vector $\hat{\mathbf{x}}^t$ is obtained as a crossover of $\mathbf{x}^t$ and a mutated vector $\bar{\mathbf{x}}^t = \mathbf{x}^{r_1} + F(\mathbf{x}^{r_3} - \mathbf{x}^{r_2})$, where $F > 0$ is a scaling factor. In what follows, we refer to this mutation operator as LIMO (linear mutation operator). For different crossover operators and many variants of DE, see, for example, Price et al (2005).

The use of perturbations, the difference vectors in this case, derived from the current population instead of utilizing an external probability distribution makes the mutation operator of DE self-adaptive (Price et al (2005)). That is, both the scale and orientation of the search are adapted to the extent of the current population. The self-adaptation in DE works especially well if all the interdependencies between decision variables in the problem are linear, but can fail to extract information of any nonlinear interdependencies between decision variables based on the relative positioning of the points in the population (Ruuska and Aittokoski (2008)).

3 New Hybrid Operator and Its Elements

In this section, we first introduce a polynomial based operator and then a new hybrid operator which utilizes both the polynomial based operator and the original LIMO operator of DE. We begin by a brief introduction to interpolation with polynomials.

3.1 Brief Background of Interpolation with Polynomials

In general, a polynomial

$$p(x) = c_d x^d + c_{d-1} x^{d-1} + \ldots + c_1 x + c_0 \tag{2}$$

can be fitted to data consisting of pairs $(x_i, y_i) \in \mathbb{R} \times \mathbb{R}$ $i = 1, \ldots, d+1$, so that it interpolates the pairs, that is, $p(x_i) = y_i$ for each i. As commonly known, the polynomial, which interpolates the given pairs and the degree of which is smaller or equal to the number of pairs plus one, is always unique (see for example Kincaid and Cheney (2002)). Here, for deriving an integral element, polynomial operator, for our hybrid

operator, we use the *Vandermonde* matrix (Kincaid and Cheney (2002)) to define coefficients c_j. This decision has been made because of its simplicity. However, the Vandermonde matrix may be ill-conditioned and therefore coefficients c_j may be inaccurate (Kincaid and Cheney (2002)). In what follows, we only consider second degree polynomials ($d = 2$ in (2)) to avoid ill-conditioned Vandermonde matrices. Another benefit of second degree polynomials is that we can express the formulas of coefficients c_j by using values x_i and y_i, $i = 1, 2, 3$, only.

For given pairs (x_1, y_1), (x_2, y_2) and (x_3, y_3), the coefficients c_2, c_1 and c_0, for a second degree polynomial p in (2), using the Vandermonde matrix are the following

$$
\begin{aligned}
c_2 &= \frac{x_3(y_2 - y_1) + x_2(y_1 - y_3) + x_1(y_3 - y_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}, \\
c_1 &= \frac{x_3^2(y_1 - y_2) + x_1^2(y_2 - y_3) + x_2^2(y_3 - y_1)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \text{ and} \\
c_0 &= \frac{x_2x_3(x_2 - x_3)y_1 + x_1x_3(x_3 - x_1)y_2 + x_1x_2(x_1 - x_2)y_3}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)}.
\end{aligned}
$$

Note that the degree of the polynomial p can be smaller than the number of the given pairs. For example, if $y_1 = y_2 = y_3$, then we get $c_2 = c_1 = 0$ and $c_0 = y_1$ (generally, this holds also for any $d > 2$ in (2), see Kincaid and Cheney (2002)).

It is well-known that interpolation methods based on polynomials are not restricted to the above type of interpolation. However, here we consider this simple approach, because we want to use a small number of data pairs. Other types of polynomial based interpolations like *splines* can be found, for example, in Kincaid and Cheney (2002) with further discussion.

3.2 Polynomial Based Point Tracking in a Higher Dimension

In this subsection, we fit a polynomial-based curve to three randomly selected decision vectors $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2 \in P$ so that the curve interpolates the vectors. In the previous subsection, polynomial p in (2) is constructed only for real values y_1, y_2 and y_3. To fit a curve to n-dimensional vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$, we use n second degree polynomials.

A *curve* $\mathbf{p}$ created for $Q = \{\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2\}$ is a function from $\mathbb{R}$ into $\mathbb{R}^n$,

$$\mathbf{p}(t) = (p_1(t), p_2(t), \ldots, p_n(t))^T, \tag{3}$$

where p_i is a polynomial from $\mathbb{R}$ into $\mathbb{R}$ for each $i = 1, \ldots, n$. Polynomials p_i, stated as

$$p_i(t) = c_2^i t^2 + c_1^i t^1 + c_0^i, \tag{4}$$

are selected so that polynomial p_i interpolates pairs $(0, x_i^0)$, $(1, x_i^1)$ and $(2, x_i^2)$ for all $i = 1, \ldots, n$. In this case, coefficients c_2, c_1 and c_0 are the following:

$$\begin{aligned} c_2^i &= \frac{x_i^0 - 2x_i^1 + x_i^2}{2}, \\ c_1^i &= \frac{4x_i^1 - 3x_i^0 - x_i^2}{2} \text{ and} \\ c_0^i &= x_i^0. \end{aligned} \tag{5}$$

3.3 A Polynomial Part for the New Operator

In the previous subsection, we have constructed a curve $\mathbf{p}$ to trace the path of $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2 \in Q$ and next we propose to use this "path"-information to generate new trial vectors. Once the polynomials $p_1, p_2, \ldots, p_n$ have been found based on Q, we get a trial vector as $\mathbf{p}(t)$ with an appropriate t-value. In the context of DE, we use $\mathbf{p}$ defined in (3)–(5) as a new POlynomial based Mutation Operator, POMO, which will be used in our hybrid operator.

As said, the POMO operator uses t as a parameter to generate new trial vectors and depending on the t-value POMO generates new trial vectors in different ways. According to the previous subsection, polynomials p_i are fitted to Q with t-values 0, 1 and 2. Once the formula for $\mathbf{p}$ as a function of t has been obtained, it can be used to generate a new trial vector with any value of t. Now, with a t-value between 0 to 2, the curve $\mathbf{p}$ interpolates and with a t-value higher than 2, the tracking curve $\mathbf{p}$ extrapolates. We limit our consideration to values between 2 and 3 because we do not want to get too far from the original vectors. In the case of interpolation, POMO produces a new trial vector between $\mathbf{x}^0$ and $\mathbf{x}^1$ or $\mathbf{x}^1$ and $\mathbf{x}^2$ depending on the selected interval. In this way, POMO tries to increase the uniform distribution

of trial vectors between $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$. To generate a new trial vector outside $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$, one can use the extrapolation version of POMO.

Let us consider three decision vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$ in the decision space as shown in Figure 1. Now, if we use LIMO at $\mathbf{x}^0$, we generate a new trial vector, denoted as $\mathbf{x}^L$ based on vectors $\mathbf{x}^1$ and $\mathbf{x}^2$. Whereas using POMO instead of LIMO, we can use the point tracking property of $\mathbf{p}$ and extract the curvature information between the decision vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$ to get vector $\mathbf{x}^i$ or $\mathbf{x}^e$. In Figure 1, vector $\mathbf{x}^i$ is produced by POMO with a t-value 1.5. In this case, vector $\mathbf{x}^i$ is between $\mathbf{x}^1$ and $\mathbf{x}^2$ as a result of the interpolation property of POMO. On the other hand, vector $\mathbf{x}^e$ has been generated with a t-value 2.5 and it aims at expanding the search area, guided by the vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$.

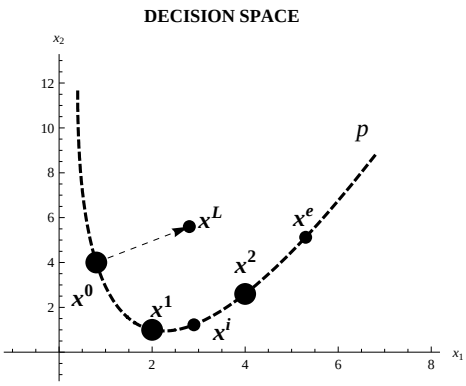

Fig. 1 Trial vector generation in the decision space using POMO and LIMO

As seen in the fitting phase of polynomials p_i, coefficients c_j require an order for the vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$. Note that vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$ are randomly selected and superscripts are here used to denote and identify the random vectors. As known, three randomly selected vectors can be ordered in six different ways and no matter what the order is, coefficients c_j can be calculated. However, it is very difficult to say beforehand which order is the best one for the polynomials p_i. For simplicity, we propose to use the order of randomly selected vectors $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$. In other words, $\mathbf{x}^0$ is the first randomly selected vector for the fitting t-value 0 and vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ for t-values 1 and 2, respectively.

3.4 New Hybrid Mutation Operator

The variable dependencies in any practical problem are a priori unknown. Here we propose a hybrid mutation operator comprising of both POMO and LIMO parts. For any DE based EMO algorithm, we suggest to use HOP so that LIMO is used with a probability P_c and POMO otherwise. Henceforth we refer to such an operator as a Hybrid OPerator (HOP). HOP is a versatile operator for an EMO algorithm because it generates trial vectors in different, yet simple ways.

As mentioned earlier, POMO can act as an extrapolation or an interpolation operator depending on the value of t. We suggest to use both extrapolation and interpolation with a prefixed probability in POMO. In what follows, the extrapolation probability is denoted by P_{extra} and the interpolation probability by P_{inter}. Naturally, we need only one of them because $P_{extra} + P_{inter} = 1$. The extrapolation operator behaves like an exploration operator and has a similar role as LIMO. The interpolation operator on the other hand exploits locally nonlinear interdependencies.

A balance of extrapolation and interpolation is essential. This can be explained with the concepts of exploration and exploitation. Although, these concepts are often utilized in the literature to describe a good optimization method, the same can be extended to POMO. In this paper, we aim at striking a balance between exploration of the decision space and exploitation of the curvature information from the sample in the decision space. POMO is constructed based on the principle of interpolation and hence it may not necessarily capture the exact nonlinear behaviour when trial vectors are generated by extrapolation. In other words, if POMO is used in HOP with a probability P_{extra} = 1.0, then it shows exploration behaviour. On the other hand, if POMO is used in HOP with a probability P_{extra} = 0, that is, P_{inter} = 1.0 meaning only interpolation, then it exploits the curvature information between chosen decision variables for trial vector generation. Using only extrapolation (exploration) can lead to slow convergence whereas performing only interpolation (exploitation) can lead to premature convergence. As a result, POMO can be trapped in locally Pareto optimal fronts. Thus, instead of having either extrapolation or in-

terpolation, one may recommend using them both. A detailed parametric study is performed in the next section for the MOEA/D algorithm to suggest a range for choosing P_{inter} (and P_{extra}).

We propose to replace the trial point generation of the LIMO operator by the hybrid operator HOP which can be described as follows:

- Generate a random number r between 0 and 1.
- If $r \leq 0.75$, set $\hat{\mathbf{x}}^t = \mathbf{x}^0 + F(\mathbf{x}^2 - \mathbf{x}^1)$
- Else set $\hat{\mathbf{x}}^t = \mathbf{p}(t)$, where t is randomly selected
  - between 0 and 2 if random number for the probability of interpolation is below P_{inter} and
  - between 2 and 3 otherwise.

Because of its simple structure, LIMO can be easily replaced with HOP in any appropriate EMO algorithm.

4 Numerical Experiments

We have proposed a new hybrid operator HOP and in this section we demonstrate how it can be used with the MOEA/D algorithm (Zhang et al (2009a)) and what advantage can be obtained with the new operator when compared to the original algorithm. We have selected MOEA/D to be used in the tests because of its superior performance in the CEC 2009 competition. However, MOEA/D is a high performance algorithm and one cannot expect drastic improvements in the performance. If we show significant statistical improvement in the performance of the MOEA/D algorithm, the strength of the new operator is supported. We first present a brief summary of the EMO algorithm used, that is, MOEA/D and subsequently present the results of the parameter study and suggestions to choose the parameter values. Finally, we present the results of numerical tests comparing HOP and the original LIMO based MOEA/D. Here, we consider all the test problems of the CEC 2007 (Huang et al (2007)) and 2009 (Zhang et al (2009b)) EMO competitions as these test problems were designed to represent various complicated problems and accepted

Table 1 Test problem instances.

Name	Number of variables	Number of objectives	Intervariable dependencies
OKA2	3	2	Nonlinear
SYMPART	30	2	Linear
S_ZDT1	30	2	Linear
S_ZDT2	30	2	Linear
S_ZDT4	30	2	Linear
S_ZDT6	30	2	Linear
R_ZDT4	10	2	Linear
S_DTLZ2_M3	30	3	Linear
S_DTLZ2_M5	30	5	Linear
S_DTLZ3_M3	30	3	Linear
S_DTLZ3_M5	30	5	Linear
R_DTLZ2_M3	30	3	Linear
R_DTLZ2_M5	30	5	Linear
WFG1_M3	24	3	Linear
WFG1_M5	28	5	Linear
WFG8_M3	24	3	Nonlinear
WFG8_M5	28	5	Nonlinear
WFG9_M3	24	3	Nonlinear
WFG9_M5	28	5	Nonlinear
UF1	30	2	Nonlinear
UF2	30	2	Nonlinear
UF3	30	2	Nonlinear
UF4	30	2	Nonlinear
UF5	30	2	Nonlinear
UF6	30	2	Nonlinear
UF7	30	2	Nonlinear
UF8	30	3	Nonlinear
UF9	30	3	Nonlinear
UF10	30	3	Nonlinear
R2_DTLZ2_M5	30	5	Linear
R2_DTLZ3_M5	30	5	Linear

for comparing different EMO algorithms. These sets contain 14 bi-objective problems, 9 problems with three and 8 problems with five objectives. In Table 1, we present the test problems with their corresponding numbers of variables and objectives and the type of dependency in their Pareto sets.

Zhang and Li (2007) proposed MOEA/D, a high performance EMO algorithm based on scalarizing functions. Subsequently, a new version of the MOEA/D algorithm was proposed in Zhang et al (2009a). The fundamental principle behind MOEA/D is that a multiobjective problem is scalarized into a number of single

objective problems with distinct weight vectors and solved simultaneously. This is because any Pareto optimal solution can be obtained by solving an appropriate scalarizing function (Miettinen (1999)). A weighted Chebyshev approach (Miettinen (1999)) is used in the MOEA/D algorithm for scalarization.

MOEA/D employs a set of N individuals and uniformly distributed weight vectors, where N is the population size. Each of these weight vectors are used to formulate a single objective problem, that is, a subproblem. A neighbourhood is defined for each individual in the population based on the distances between the weight vectors. Genetic operations (i.e., selection, crossover and mutation) are performed in each of the neighbourhoods. An offspring is generated by crossover and mutation with randomly selected parents in the neighbourhood. This offspring is then compared with the parents and each of the neighbours based on the value of the scalarizing function. If the offspring is better than any of the individuals used in the comparison, it replaces them. The algorithm finally terminates when a suitable termination criterion is met. MOEA/D used in this paper uses dynamic resource allocation to assign different amounts of computational effort to different subproblems. A complete description of the algorithm is provided in Zhang et al (2009a).

The MOEA/D algorithm employs a linear DE mutation operator LIMO as a crossover operator. The offspring generated by crossover is subsequently mutated using a mutation operator (Zhang et al (2009a)) with a probability to produce a new offspring. For our comparisons we change the crossover operator of MOEA/D and, instead, employ HOP and the rest of the algorithm remains unchanged. In this paper, we refer to the modified MOEA/D algorithm with HOP as MOEA/D-HOP. As described in the previous section, inside MOEA/D-HOP, we have a probability for using either POMO or LIMO.

4.1 Test Setting

The MOEA/D algorithm as suggested in Zhang et al (2009a) is used as the EMO algorithm in the comparisons with MOEA/D-HOP. The program for the MOEA/D algorithm provided in *http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm* is used

for tests. The test problems include all the 31 box-constrained test problems of the CEC 2007/2009 competitions. To have a fair comparison with the competition results, we borrow the performance metric and parameter settings of the CEC 2009 competition. Next, we summarize the parameter setting used for all subsequent tests:

(1) Performance metric: Inverted generational distance (IGD) is used as the performance metric. If P^* is the set of uniformly distributed Pareto optimal solutions in the objective space and P the obtained approximation set of non-dominated solutions in the objective space from the EMO algorithm, the IGD value for the approximation set is defined by:

$$IGD(P, P^*) = \frac{\sum_{\mathbf{v} \in P^*} d(\mathbf{v}, P)}{|P^*|} \tag{6}$$

where $d(\mathbf{v}, P)$ is the minimum Euclidean distance between $\mathbf{v}$ and points in P and $|P^*|$ is the number of points in P^*.

(2) The maximal number of approximate solutions ($|P|$) for the IGD value calculation: 100 for bi-objective problems, 150 for three objectives, and 800 for five objectives.

(3) Algorithm stopping criterion is the maximal number of function evaluations = $300,000$.

(4) The number of independent runs for statistical tests = 30.

In what follows, we present the parameter settings specific to the MOEA/D algorithm. We want to point out that the parameter settings for both MOEA/D and MOEA/D-HOP are the same as in Zhang et al (2009a).

(1) Population size (N): 600 for bi-objective problems, 1000 for three objectives, and 1500 for five objectives.

(2) Number of weight vectors in the neighbourhood of each weight vector (T) = $0.1N$ and n_r = $0.01N$.

(3) Probability for selection of mating/update range δ = 0.9.

(4) Crossover ratio (CR) = 1.0 and scale factor (F) = 0.5. Mutation probability P_m = $1/n$ and mutation distribution index η = 20.

The parameters for HOP are the following.

(1) Probability of LIMO operator P_c = 0.75 (heuristically set based on experiments).
(2) Probability of interpolation P_{inter} and extrapolation ($P_{extra} = 1 - P_{inter}$) set based on an experimental study, which is described later in this section.
(3) Parameter t chosen randomly in the interval $0 - 2$ for interpolation and $2 - 3$ for extrapolation.

It can be seen from the above parameter setting that we may have a larger number of individuals in the final population (Π) than allowed for the calculation of IGD values. Hence, we prune the population Π to match the competition specifications using the procedure proposed in Zhang et al (2009a) and obtain the final approximation set P.

For each test problem, when comparing the MOEA/D and the MOEA/D-HOP algorithms, we record the best, median and worst IGD values from 30 independent runs (note that the smaller the IGD value the better). The IGD values of both algorithms are subsequently used to perform the Wilcoxon rank sum test (Gibbons and Chakraborti (2003)), a non-parametric statistical test at 5% significance level. The median IGD value of the MOEA/D algorithm is used as the reference value for the final conclusion to be significant success or significant failure of MOEA/D-HOP against MOEA/D. The *p-value* is the probability of observing the given sample result under the assumption that the null hypothesis is true. The null hypothesis here is defined as $H_o : A = B$, where A and B are the median IGD values of the MOEA/D and MOEA/D-HOP algorithms, respectively. The Wilcoxon rank sum test is used separately for each test problem to determine whether the perceived difference in the performance levels is statistically significant or not. A significant test result is declared a significant success, if the median IGD value is less than the reference value, and a significant failure, if the median IGD value is greater than the reference value. The overall performance of the MOEA/D-HOP algorithm is judged based on the total number of significant successes achieved over the MOEA/D algorithm.

4.2 Test Results

The POMO part in the HOP is used for interpolation with a fixed probability (P_{inter}) and extrapolation otherwise. Statistical tests were performed with MOEA/D-HOP and P_{inter} equal to 0.00, 0.25, 0.50, 0.75 and 1.00 for all test problems. For each value of P_{inter}, we count the number of significant successes N_{win}. The particular P_{inter} for which N_{win} is maximum, is considered as an ideal setting for P_{inter}. In tables 2, 3, 4, 5 and 6 we summarize the test results obtained for different values of P_{inter}. They contain best, median and worst IGD values for both MOEA/D and MOEA/D-HOP and the *p-values* for all 31 test problems. The significant success of any algorithm by the Wilcoxon rank sum test is marked in **bold face**.

In Table 2, the P_{inter} value is 0.00; in other words we only extrapolate to generate trial vectors in the POMO part. It can be observed that the numbers of significant successes of MOEA/D and MOEA/D-HOP are 10 and 6, respectively (denoted in **bold face**). The IGD value difference between the algorithms is insignificant for 15 test problems.

In what follows, we consider using interpolation in POMO as well. The results for P_{inter} = 0.25 are shown in Table 3. A *p-value* greater than 5% significance level indicating insignificance is observed in 19 test problems and the number of significant failures of the MOEA/D-HOP reduces to 5 as against 10 in Table 2. This shows the importance of interpolation in POMO. However, the IGD values using this parameter setting do not yet show any significant performance enhancement for MOEA/D-HOP algorithm. Next, we further increase the value of P_{inter} to 0.5. In this setting, the extrapolation and interpolation in POMO is used with equal probability. The results in Table 4 show further increase in the number of significant successes for the MOEA/D-HOP algorithm to be 12 and insignificant performance improvements can be seen in 14 test problems.

Next, P_{inter} is increased to 0.75 and now we have a higher probability of performing interpolation in POMO. In such a test setting, the number of significant successes of the MOEA/D-HOP increases to 14 and the number of insignificant performance improvements reduces to 12, as shown in Table 5. One can also see

that when P_{inter} values are 0.25, 0.50 and 0.75, the number of significant failures for MOEA/D-HOP remains 5. Hence, with the increase in interpolation, the number of significant successes increases. Finally, when the P_{inter} value is 1.00, in other words, when we only interpolate to generate trial vectors in the POMO part, the number of significant successes of MOEA/D-HOP decreases to 9 as can be seen in Table 6. Also the number of significant failures in MOEA/D-HOP increases to 7. Thus, it can be concluded from the studies that the extremes of only extrapolation or interpolation in POMO degrade the performance of the MOEA/D-HOP algorithm and both extrapolation and interpolation are necessary for the best performance of MOEA/D-HOP. Based on the experiments, we can suggest the value of P_{inter} to be 0.75.

Furthermore, for curiosity we repeat the tests with P_{inter} = 0.67. From Table 7 we can observe that the number of significant successes for the MOEA/D-HOP algorithm is 13. Thus, the performance of MOEA/D-HOP has not much deteriorated. Hence, the algorithm is rather stable for any value of P_{inter} between 0.50 and 0.75. Henceforth, for further discussions and conclusions we consider only Table 5 (with P_{inter} = 0.75).

It can be seen from Table 5 that MOEA/D-HOP performs equally well on problems with linear and nonlinear interdependencies between decision variables. Typically, MOEA/D-HOP either performs better than or on par with the LIMO based MOEA/D algorithm, which provides us confidence for using the hybrid based operator HOP. One can say that by employing HOP, the solutions typically do not get worse than the original ones. In the case of more than two objectives, it can be observed from Table 5 that MOEA/D-HOP portrays a better performance when compared to MOEA/D. Even in the case of the worst IGD values of 30 runs, the IGD values of the MOEA/D-HOP algorithm are better in 22 of 31 test problems when compared to the MOEA/D algorithm. This shows that the HOP based algorithm can be a safe and more reliable choice for solving multiobjective optimization problems.

The MOEA/D algorithm with the LIMO operator showed a bad performance in the UF2, UF4, UF5, UF9 and UF10 test problems against some of its competitors in

the CEC 2009 competition. Incidentally, the MOEA/D-HOP algorithm shows significantly improved performance in the UF2, UF4 and UF5 test problems and insignificant improvement in the UF9 and UF10 test problems. On the other hand, with the MOEA/D-HOP algorithm, the median IGD value of the UF2 test problem is 0.0060, which is even better than the best in the competition (0.00615). It must be noted that UF2 has a nonlinear Pareto set and for this reason it may be very difficult for any linear operator to generate trial vectors so that the entire Pareto set could be covered. This observation supports our idea of the importance of having operators that are robust and not linear.

In Figure 2, we show the nondominated sets of the UF2 test problem representing the median IGD values for both the MOEA/D-HOP and MOEA/D algorithms. Both algorithms seem to provide a good spread of solutions on the Pareto front. However, on a close inspection one can spot a hole in the nondominated set produced by the MOEA/D algorithm. The LIMO operator has failed to generate any trial vectors in that region. A magnified illustration of this hole is also shown in Figure 2. It can be seen that the nondominated set of the MOEA/D-HOP algorithm has no holes and the HOP operator has generated trial vectors in the whole region.

In addition, we also depict the nondominated sets corresponding to the worst IGD values for both the MOEA/D and MOEA/D-HOP algorithms for the UF2 test problem in Figure 3. The plot shows a much smaller discontinuity in the nondominated set of the MOEA/D-HOP algorithm when compared to the MOEA/D algorithm. Figures 2 and 3 demonstrate the reliability of the MOEA/D-HOP algorithm in handling Pareto sets as compared to the MOEA/D algorithm.

5 Conclusions and Future Research Directions

In this paper, we have developed a new hybrid operator HOP consisting of a standard mutation operator LIMO of DE and a polynomial mutation operator POMO. With these two elements included, we have constructed a mechanism which uses the curvature information based on presently known vectors in the calculations of

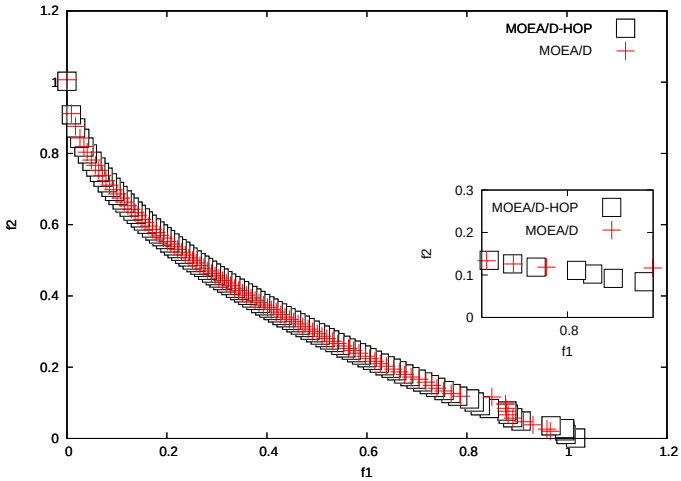

Fig. 2 Nondominated sets of the UF2 test problem for MOEA/D and MOEA/D-HOP representing median IGD values.

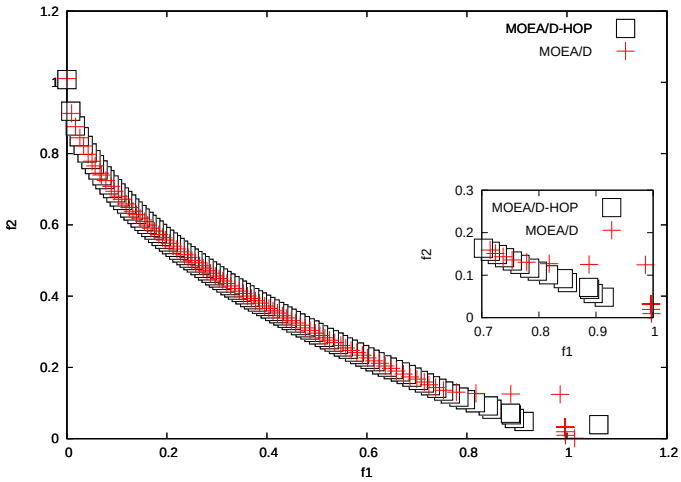

Fig. 3 Nondominated sets of the UF2 test problem for MOEA/D and MOEA/D-HOP representing worst IGD values.

new trial vectors and the resulting operator is robust for different types of problems. Yet, the operator proposed is simple to implement.

In HOP, we have proposed to use both POMO and LIMO operators and both interpolation and extrapolation inside POMO for trial vector generation. We demonstrated the efficacy of the new operator with comparisons to MOEA/D, a winning

algorithm of a recent CEC competition, which uses the linear operator LIMO. We have observed significant improvement with the new operator in the performance of the MOEA/D algorithm in various types of problems as against the original MOEA/D algorithm. In spite of the fact that one could not expect drastic improvements in the performance of MOEA/D, because it is a high performance algorithm, the new operator provided robust performance and better results for many test problems.

In the future research, the choice of a proper t-value can be further considered including a possibility of ordering the vectors used in the construction phase of the polynomial. In addition, a self-adaptive way of using both LIMO and POMO operators for multiobjective optimization may be formulated instead of a pre-fixed probability. By adaptively using different operators we can better handle demanding real-life multiobjective optimization problems for which we typically do not know the properties of the Pareto optimal set in advance. The performance of the operator with different EMO algorithms is also in our future scopes of study.

Acknowledgements The authors would like to thank Prof. Raino A. E. Mäkinen and Dr. Timo Aittokoski for fruitful discussions.

References

Abbass HA (2002) The self-adaptive Pareto differential evolution algorithm. In: Proceedings of the Congress on Evolutionary Computation (CEC '02), IEEE Press, pp 831–836

Ali M, Pant M (2010) Improving the performance of differential evolution algorithm using cauchy mutation. Soft Computing DOI 10.1007/s00500-010-0655-2

Ali MM, Törn A, Viitanen S (1997) A numerical comparison of some modified controlled random search algorithms. Journal of Global Optimization 11(4):377–385

Babu BV, Jehan MML (2003) Differential evolution for multi-objective optimization. In: Proceedings of the Congress on Evolutionary Computation (CEC '03), IEEE Press, pp 2696–2073

Bazaraa MS, Sherali HD, Shetty CM (2006) Nonlinear Programming: Theory and Algorithms. John Wiley and Sons, Hoboken

Coello CAC, Lamont GB, Veldhuizen DAV (2007) Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, New York

Deb K (2001) Multi-objective Optimization using Evolutionary Algorithms. Wiley, Chichester

Fan HY, Lampinen J (2003) A trigonometric mutation operation to differential evolution. Journal of Global Optimization 27:105–129

Gibbons JD, Chakraborti S (2003) Nonparametric Statistical Inference. Marcel Dekker, New York

Huang VL, Qin AQ, Deb K, Zitzler E, Suganthan PN, Liang JJ, Preuss M, Huband S (2007) Problem definitions for performance assessment of multi-objective optimization algorithms. Tech. rep., Nanyang Technological University

Jin Y, Sendhoff B (2003) Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In: Proceedings of the Congress on Evolutionary Computation (CEC '03), IEEE Press, pp 1910–1917

Kaelo P, Ali MM (2007) Differential evolution algorithms using hybrid mutation. Computational Optimization and Applications 5(2):231–246

Kincaid D, Cheney W (2002) Numerical Analysis: Mathematics of Scientific Computing. Brooks/Cole Publishing Co.

Kukkonen S, Lampinen J (2005) GDE3: The third evolution step of generalized differential evolution. In: Proceedings of the Congress on Evolutionary Computation (CEC '05), IEEE Press, pp 443–450

Kukkonen S, Lampinen J (2006) Constrained real-parameter optimization with generalized differential evolution. In: Proceedings of the Congress on Evolutionary Computation (CEC '06), IEEE Press, pp 207–214

Lara A, Sanchez G, Coello CAC, Schütze O (2010) HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 14(1):112–132

Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. Soft Computing 9:448–462

Miettinen K (1999) Nonlinear Multiobjective Optimization. Kluwer, Boston

Nocedal J, Wright SJ (1999) Numerical Optimization. Springer, New York

Okabe T, Jin Y, Olhofer M, Sendhoff B (2004) On test functions for evolutionary multi-objective optimization. In: Yao X, et al (eds) Proceedings of the Parallel Problem Solving from Nature (PPSN VIII 2004), Springer-Verlag, pp 792–802

Price KV, Storn RM, Lampinen JA (2005) Differential Evolution - A Practical Approach to Global Optimization. Springer-Verlag, Berlin/Heidelberg

Robic T, Filipic B (2005) DEMO: Differential evolution for multiobjective optimization. In: Coello CAC, et al (eds) Proceedings of the Evolutionary Multi-criterion Optimization (EMO 2005), Springer, pp 520–533

Ruuska S, Aittokoski T (2008) The effect of trial point generation schemes on the efficiency of population-based global optimization algorithms. In: Proceedings of the International Conference on Engineering Optimization (EngOpt '08)

Santana-Quintero LV, Coello CAC (2005) An algorithm based on Differential Evolution for multi-objective problems. International Journal of Computational Intelligence Research 1(2):151–169

Schütze O, Talbi E, Coello CC, Santana-Quintero LV, Pulido GT (2007) A memetic PSO algorithm for scalar optimization problems. In: Proceedings of the Symposium on Swarm Intelligence (SIS '07), IEEE Press, pp 128–134

Storn R, Price K (1996) Minimizing the real functions of the ICEC'96 contest by Differential Evolution. In: Proceedings of the Conference on Evolutionary Computation (ICEC '96), IEEE Press, pp 842–844

Wang X, Hao M, Cheng Y, Lei R (2009) PDE-PEDA: A new Pareto-based multi-objective optimization algorithm. Journal of Universal Computer Science 15(4):722–741

Zhang Q, Li H (2007) MOEA/D: A multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6):712–731

Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 12(1):41–63

Zhang Q, Liu W, Li H (2009a) The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proceedings of the Conference on Evolutionary Computation (CEC '09), IEEE Press, pp 203–208

Zhang Q, Zhou A, Zhao SZ, Suganthan PN, Liu W, Tiwari S (2009b) Multiobjective optimization test instances for the CEC09 special session and competition. Tech. Rep. CES-487, University of Essex and Nanyang Technological University

Table 2 MOEA/D-HOP with 0 % interpolation and 100 % extrapolation.

Problem	MOEA/D			MOEA/D-HOP			*p*-value
	Best	Median	Worst	Best	Median	Worst	
OKA2	0.0295	0.0295	0.0295	0.0295	0.0295	0.0295	0.5325
SYMPART	0.0208	0.0220	0.0230	0.0214	0.0222	0.0238	0.08503
S_ZDT1	0.0051	**0.0058**	0.0066	0.0060	0.0068	0.0080	1.572e-13
S_ZDT2	0.0059	**0.0067**	0.0081	0.0061	0.0080	0.0097	5.868e-09
S_ZDT4	0.0041	**0.0059**	0.0082	0.0053	0.0099	0.0286	1.086e-10
S_ZDT6	0.8429	**0.9343**	1.0557	0.9107	1.0249	1.1463	1.537e-08
R_ZDT4	0.2554	0.6674	2.8821	0.1254	1.2737	3.0742	0.2359
S_DTLZ2_M3	0.0414	0.0424	0.0434	0.0416	0.0426	0.0437	0.1774
S_DTLZ2_M5	0.1047	0.1076	0.1099	0.1042	0.1069	0.1096	0.1026
S_DTLZ3_M3	14.3940	50.2210	121.6451	12.8652	51.2645	135.5391	0.6973
S_DTLZ3_M5	39.6762	120.7275	214.8927	45.7423	121.1439	257.6923	0.7747
R_DTLZ2_M3	37.4759	73.8879	157.8043	34.5266	84.4257	163.8599	0.2996
R_DTLZ2_M5	40.9777	94.2054	190.5095	44.4605	105.5567	162.8698	0.4853
WFG1_M3	1.3605	1.3691	1.3798	1.3285	**1.3495**	1.3664	3.298e-15
WFG1_M5	1.8004	1.8235	1.8989	1.7949	**1.8035**	1.8142	1.25e-09
WFG8_M3	0.2140	**0.2194**	0.2301	0.2166	0.2256	0.2351	1.714e-06
WFG8_M5	0.8839	**0.9475**	0.9936	0.8986	0.9639	1.0050	0.006969
WFG9_M3	0.1633	0.1721	0.2447	0.1628	0.1787	0.2296	0.1194
WFG9_M5	0.8734	0.9202	1.0489	0.8775	0.9215	1.0123	0.4063
UF1	0.0043	**0.0046**	0.0054	0.0047	0.0051	0.0098	3.632e-12
UF2	0.0053	0.0068	0.0109	0.0055	**0.0059**	0.0066	0.001092
UF3	0.0040	0.0050	0.0737	0.0042	0.0047	0.0112	0.3504
UF4	0.0522	0.0601	0.0721	0.0506	**0.0560**	0.0662	3.775e-05
UF5	0.0643	0.1704	0.3501	0.0870	**0.1385**	0.3369	0.02738
UF6	0.0037	0.0061	0.0113	0.0029	0.0062	0.0102	0.3354
UF7	0.0040	**0.0042**	0.0236	0.0053	0.0071	0.0102	5.959e-08
UF8	0.0510	0.0581	0.0629	0.0497	**0.0551**	0.0641	0.0199
UF9	0.0354	**0.0399**	0.1514	0.0387	0.0442	0.1514	0.0055
UF10	0.3531	0.4449	0.6221	0.3814	0.4485	0.5978	0.6865
R2_DTLZ2_M5	0.1093	**0.1117**	0.1143	0.1087	0.1143	0.1218	0.001366
R2_DTLZ3_M5	78.9882	140.9533	246.1105	66.3643	130.9731	207.3482	0.398

Table 3 MOEA/D-HOP with 25 % interpolation and 75 % extrapolation.

Problem	MOEA/D			MOEA/D-HOP			*p*-value
	Best	Median	Worst	Best	Median	Worst	
OKA2	0.0295	0.0295	0.0295	0.0295	0.0295	0.0295	0.1681
SYMPART	0.0208	0.0220	0.0230	0.0211	0.0220	0.0243	0.959
S_ZDT1	0.0051	**0.0058**	0.0066	0.0055	0.0060	0.0068	0.002738
S_ZDT2	0.0059	0.0067	0.0081	0.0057	0.0068	0.0078	0.2188
S_ZDT4	0.0041	**0.0059**	0.0082	0.0052	0.0080	0.0154	4.473e-06
S_ZDT6	0.8429	**0.9343**	1.0557	0.8111	1.0247	1.1138	7.739e-05
R_ZDT4	0.2554	0.6674	2.8821	0.0039	0.8859	1.9510	0.6333
S_DTLZ2_M3	0.0414	0.0424	0.0434	0.0414	0.0422	0.0431	0.06112
S_DTLZ2_M5	0.1047	0.1076	0.1099	0.1028	**0.1055**	0.1090	7.398e-08
S_DTLZ3_M3	14.3940	50.2210	121.6451	13.0816	39.1385	107.9345	0.1229
S_DTLZ3_M5	39.6762	120.7275	214.8927	54.8694	103.1090	189.4424	0.2665
R_DTLZ2_M3	37.4759	73.8879	157.8043	23.3622	81.1145	140.1990	0.6865
R_DTLZ2_M5	40.9777	94.2054	190.5095	40.4856	93.7841	158.6842	0.8545
WFG1_M3	1.3605	1.3691	1.3798	1.3212	**1.3364**	1.3521	1.691e-17
WFG1_M5	1.8004	1.8235	1.8989	1.7916	**1.8048**	1.8448	6.198e-07
WFG8_M3	0.2140	0.2194	0.2301	0.2039	0.2217	0.2308	0.2244
WFG8_M5	0.8839	0.9475	0.9936	0.8960	0.9424	1.0070	0.7191
WFG9_M3	0.1633	0.1721	0.2447	0.1634	0.1752	0.2344	0.1421
WFG9_M5	0.8734	0.9202	1.0489	0.8701	0.9410	1.0494	0.5325
UF1	0.0043	**0.0046**	0.0054	0.0045	0.0052	0.0100	3.556e-09
UF2	0.0053	0.0068	0.0109	0.0051	**0.0058**	0.0067	4.7e-05
UF3	0.0040	0.0050	0.0737	0.0041	0.0074	0.0256	0.8315
UF4	0.0522	0.0601	0.0721	0.0504	**0.0559**	0.0633	0.0001252
UF5	0.0643	0.1704	0.3501	0.0864	**0.1269**	0.3976	0.0105
UF6	0.0037	0.0061	0.0113	0.0032	0.0063	0.0089	0.4946
UF7	0.0040	**0.0042**	0.0236	0.0049	0.0059	0.0083	2.571e-07
UF8	0.0510	0.0581	0.0629	0.0497	**0.0547**	0.0641	0.0001433
UF9	0.0354	0.0399	0.1514	0.0353	0.0937	0.1566	0.3208
UF10	0.3531	0.4449	0.6221	0.3725	0.4420	0.5943	0.7747
R2_DTLZ2_M5	0.1093	0.1117	0.1143	0.1070	0.1112	0.1211	0.552
R2_DTLZ3_M5	78.9882	140.9533	246.1105	37.6952	127.3110	231.3682	0.4404

Table 4 MOEA/D-HOP with 50 % interpolation and 50 % extrapolation.

Problem	MOEA/D			MOEA/D-HOP			*p*-value
	Best	Median	Worst	Best	Median	Worst	
OKA2	0.0295	0.0295	0.0295	0.0295	0.0295	0.0295	0.1342
SYMPART	0.0208	0.0220	0.0230	0.0214	0.0219	0.0227	0.2418
S_ZDT1	0.0051	0.0058	0.0066	0.0048	**0.0055**	0.0063	0.0008203
S_ZDT2	0.0059	0.0067	0.0081	0.0054	**0.0062**	0.0074	0.0008203
S_ZDT4	0.0041	**0.0059**	0.0082	0.0054	0.0082	0.0253	1.517e-05
S_ZDT6	0.8429	**0.9343**	1.0557	0.8487	0.9728	1.1046	0.01425
R_ZDT4	0.2554	0.6674	2.8821	0.1253	0.7395	2.3153	0.4063
S_DTLZ2_M3	0.0414	0.0424	0.0434	0.0415	0.0422	0.0435	0.2359
S_DTLZ2_M5	0.1047	0.1076	0.1099	0.1013	**0.1042**	0.1076	7.957e-11
S_DTLZ3_M3	14.3940	50.2210	121.6451	7.9545	**33.0329**	82.1285	0.001525
S_DTLZ3_M5	39.6762	120.7275	214.8927	38.4916	108.2110	196.5416	0.1973
R_DTLZ2_M3	37.4759	73.8879	157.8043	35.4879	80.3239	174.2425	0.4492
R_DTLZ2_M5	40.9777	94.2054	190.5095	36.9194	88.6641	152.4668	0.5039
WFG1_M3	1.3605	1.3691	1.3798	1.3144	**1.3291**	1.3518	1.691e-17
WFG1_M5	1.8004	1.8235	1.8989	1.7952	**1.8085**	1.8394	1.641e-05
WFG8_M3	0.2140	0.2194	0.2301	0.2106	0.2209	0.2365	0.398
WFG8_M5	0.8839	0.9475	0.9936	0.8846	**0.9278**	0.9728	0.02434
WFG9_M3	0.1633	0.1721	0.2447	0.1643	0.1749	0.1860	0.3504
WFG9_M5	0.8734	**0.9202**	1.0489	0.8884	0.9726	1.0782	0.008008
UF1	0.0043	**0.0046**	0.0054	0.0045	0.0050	0.0084	2.736e-08
UF2	0.0053	0.0068	0.0109	0.0051	**0.0060**	0.0069	0.001525
UF3	0.0040	0.0050	0.0737	0.0041	0.0084	0.0220	0.6865
UF4	0.0522	0.0601	0.0721	0.0508	**0.0564**	0.0632	0.0004003
UF5	0.0643	0.1704	0.3501	0.0739	**0.1207**	0.2220	0.001795
UF6	0.0037	0.0061	0.0113	0.0031	0.0062	0.0095	0.5422
UF7	0.0040	**0.0042**	0.0236	0.0049	0.0060	0.0082	3.139e-07
UF8	0.0510	0.0581	0.0629	0.0489	**0.0549**	0.0614	1.403e-05
UF9	0.0354	0.0399	0.1514	0.0361	0.0414	0.1529	0.3738
UF10	0.3531	0.4449	0.6221	0.3574	0.4349	0.5391	0.7524
R2_DTLZ2_M5	0.1093	0.1117	0.1143	0.1065	**0.1092**	0.1162	0.0006116
R2_DTLZ3_M5	78.9882	140.9533	246.1105	59.5222	131.0013	251.4273	0.07228

Table 5 MOEA/D-HOP with 75 % interpolation and 25 % extrapolation.

Problem	MOEA/D			MOEA/D-HOP			*p*-value
	Best	Median	Worst	Best	Median	Worst	
OKA2	0.0295	**0.0295**	0.0295	0.0295	0.0295	0.0295	0.02158
SYMPART	0.0208	0.0220	0.0230	0.0211	0.0218	0.0235	0.3898
S_ZDT1	0.0051	0.0058	0.0066	0.0048	**0.0052**	0.0057	4.757e-10
S_ZDT2	0.0059	0.0067	0.0081	0.0055	**0.0061**	0.0069	1.193e-06
S_ZDT4	0.0041	**0.0059**	0.0082	0.0050	0.0070	0.0147	0.0055
S_ZDT6	0.8429	0.9343	1.0557	0.8608	0.9574	1.0287	0.4492
R_ZDT4	0.2554	0.6674	2.8821	0.1253	0.5265	2.3153	0.08778
S_DTLZ2_M3	0.0414	0.0424	0.0434	0.0411	**0.0418**	0.0432	0.005242
S_DTLZ2_M5	0.1047	0.1076	0.1099	0.1017	**0.1038**	0.1058	2.029e-16
S_DTLZ3_M3	14.3940	50.2210	121.6451	3.3976	**30.4582**	50.0858	3.773e-06
S_DTLZ3_M5	39.6762	120.7275	214.8927	30.7891	**96.5393**	149.8983	0.008774
R_DTLZ2_M3	37.4759	73.8879	157.8043	23.5514	60.8510	179.0502	0.1421
R_DTLZ2_M5	40.9777	94.2054	190.5095	39.4594	77.5597	167.7533	0.06763
WFG1_M3	1.3605	1.3691	1.3798	1.3041	**1.3254**	1.3591	1.691e-17
WFG1_M5	1.8004	1.8235	1.8989	1.7982	**1.8108**	1.8724	0.00577
WFG8_M3	0.2140	0.2194	0.2301	0.2093	0.2175	0.2322	0.07718
WFG8_M5	0.8839	0.9475	0.9936	0.8677	**0.9156**	0.9797	0.0004525
WFG9_M3	0.1633	0.1721	0.2447	0.1612	0.1762	0.1889	0.1591
WFG9_M5	0.8734	0.9202	1.0489	0.8662	0.9585	1.1233	0.08778
UF1	0.0043	**0.0046**	0.0054	0.0045	0.0053	0.0088	3.835e-08
UF2	0.0053	0.0068	0.0109	0.0049	**0.0060**	0.0078	0.0105
UF3	0.0040	**0.0050**	0.0737	0.0042	0.0103	0.0273	0.03445
UF4	0.0522	0.0601	0.0721	0.0524	**0.0582**	0.0646	0.004313
UF5	0.0643	0.1704	0.3501	0.0841	**0.1140**	0.2188	0.0009748
UF6	0.0037	0.0061	0.0113	0.0032	0.0055	0.0086	0.2301
UF7	0.0040	**0.0042**	0.0236	0.0046	0.0060	0.0111	3.465e-07
UF8	0.0510	0.0581	0.0629	0.0498	**0.0551**	0.0611	0.001998
UF9	0.0354	0.0399	0.1514	0.0351	0.0397	0.1577	0.9707
UF10	0.3531	0.4449	0.6221	0.3343	0.4214	0.5734	0.1922
R2_DTLZ2_M5	0.1093	0.1117	0.1143	0.1055	**0.1104**	0.1191	0.03445
R2_DTLZ3_M5	78.9882	140.9533	246.1105	51.8876	121.9305	205.0172	0.1462

Table 6 MOEA/D-HOP with 100 % interpolation and 0 % extrapolation.

Problem	MOEA/D			MOEA/D-HOP			*p*-value
	Best	Median	Worst	Best	Median	Worst	
OKA2	0.0295	0.0295	0.0295	0.0295	0.0295	0.0295	0.1727
SYMPART	0.0208	0.0220	0.0230	0.0212	0.0218	0.0227	0.1681
S_ZDT1	0.0051	0.0058	0.0066	0.0046	**0.0050**	0.0053	4.6e-15
S_ZDT2	0.0059	0.0067	0.0081	0.0050	**0.0059**	0.0067	9.536e-10
S_ZDT4	0.0041	**0.0059**	0.0082	0.0051	0.0084	0.0126	4.215e-07
S_ZDT6	0.8429	**0.9343**	1.0557	0.8227	0.9935	1.0938	0.002468
R_ZDT4	0.2554	0.6674	2.8821	0.1253	0.6692	3.8605	0.2478
S_DTLZ2_M3	0.0414	0.0424	0.0434	0.0410	**0.0419**	0.0426	0.0002269
S_DTLZ2_M5	0.1047	0.1076	0.1099	0.1012	**0.1036**	0.1063	1.133e-15
S_DTLZ3_M3	14.3940	50.2210	121.6451	12.6367	**29.3421**	67.2494	0.000117
S_DTLZ3_M5	39.6762	120.7275	214.8927	39.6001	**77.6292**	140.6771	2.6e-05
R_DTLZ2_M3	37.4759	73.8879	157.8043	28.7169	65.7904	150.3852	0.6022
R_DTLZ2_M5	40.9777	94.2054	190.5095	37.1947	90.3441	145.6121	0.3429
WFG1_M3	1.3605	1.3691	1.3798	1.3030	**1.3247**	1.3428	1.691e-17
WFG1_M5	1.8004	1.8235	1.8989	1.7913	1.8220	1.8954	0.5819
WFG8_M3	0.2140	0.2194	0.2301	0.2091	0.2198	0.2313	0.592
WFG8_M5	0.8839	0.9475	0.9936	0.8354	**0.9122**	1.0168	0.0001993
WFG9_M3	0.1633	0.1721	0.2447	0.1655	0.1741	0.1872	0.6865
WFG9_M5	0.8734	**0.9202**	1.0489	0.8948	1.0156	1.1393	7.739e-05
UF1	0.0043	**0.0046**	0.0054	0.0045	0.0073	0.0998	5.273e-12
UF2	0.0053	**0.0068**	0.0109	0.0074	0.0355	0.0722	6.334e-12
UF3	0.0040	**0.0050**	0.0737	0.0048	0.0147	0.0289	0.001092
UF4	0.0522	0.0601	0.0721	0.0577	0.0609	0.0699	0.9124
UF5	0.0643	0.1704	0.3501	0.0938	0.1816	0.3214	0.4946
UF6	0.0037	0.0061	0.0113	0.0033	0.0061	0.1234	0.4063
UF7	0.0040	**0.0042**	0.0236	0.0042	0.0058	0.0120	6.198e-07
UF8	0.0510	0.0581	0.0629	0.0509	0.0578	0.0772	0.3898
UF9	0.0354	0.0399	0.1514	0.0344	0.0918	0.1551	0.9357
UF10	0.3531	0.4449	0.6221	0.3447	0.4358	0.5609	0.4581
R2_DTLZ2_M5	0.1093	0.1117	0.1143	0.1053	**0.1087**	0.1149	6.724e-05
R2_DTLZ3_M5	78.9882	140.9533	246.1105	63.8735	128.7742	254.2617	0.1591

Table 7 MOEA/D-HOP with 67 % interpolation and 33 % extrapolation.

Problem	MOEAD			MOEAD + POMO			*p*-value
	Best	Median	Worst	Best	Median	Worst	
OKA2	0.0295	**0.0295**	0.0295	0.0295	0.0295	0.0295	0.03853
SYMPART	0.0208	0.0220	0.0230	0.0212	0.0220	0.0227	0.6125
S_ZDT1	0.0051	0.0058	0.0066	0.0047	**0.0053**	0.0058	2.736e-08
S_ZDT2	0.0059	0.0067	0.0081	0.0054	**0.0062**	0.0072	0.0002269
S_ZDT4	0.0041	**0.0059**	0.0082	0.0056	0.0085	0.0202	1.214e-08
S_ZDT6	0.8429	**0.9343**	1.0557	0.8714	0.9718	1.0694	0.0191
R_ZDT4	0.2554	0.6674	2.8821	0.1253	0.5266	2.3153	0.08778
S_DTLZ2_M3	0.0414	0.0424	0.0434	0.0411	0.0422	0.0429	0.0995
S_DTLZ2_M5	0.1047	0.1076	0.1099	0.1010	**0.1038**	0.1075	4.842e-13
S_DTLZ3_M3	14.3940	50.2210	121.6451	6.3860	**33.4231**	66.7001	0.0001252
S_DTLZ3_M5	39.6762	120.7275	214.8927	22.7876	**89.9467**	150.9804	0.00336
R_DTLZ2_M3	37.4759	73.8879	157.8043	28.5754	76.8656	142.8227	0.7635
R_DTLZ2_M5	40.9777	94.2054	190.5095	40.0495	79.7125	125.1337	0.1159
WFG1_M3	1.3605	1.3691	1.3798	1.3006	**1.3237**	1.3555	1.691e-17
WFG1_M5	1.8004	1.8235	1.8989	1.7935	**1.8113**	1.8457	0.001222
WFG8_M3	0.2140	0.2194	0.2301	0.2094	0.2216	0.2347	0.3659
WFG8_M5	0.8839	0.9475	0.9936	0.8560	**0.9293**	0.9994	0.04146
WFG9_M3	0.1633	0.1721	0.2447	0.1602	0.1752	0.1872	0.1091
WFG9_M5	0.8734	**0.9202**	1.0489	0.8700	0.9794	1.0348	0.01004
UF1	0.0043	**0.0046**	0.0054	0.0045	0.0055	0.0089	6.798e-11
UF2	0.0053	0.0068	0.0109	0.0050	**0.0060**	0.0092	0.01618
UF3	0.0040	0.0050	0.0737	0.0041	0.0051	0.0236	0.7191
UF4	0.0522	0.0601	0.0721	0.0509	**0.0564**	0.0633	2.233e-05
UF5	0.0643	0.1704	0.3501	0.0599	**0.1174**	0.2931	0.000258
UF6	0.0037	0.0061	0.0113	0.0030	0.0060	0.0091	0.9707
UF7	0.0040	**0.0042**	0.0236	0.0049	0.0056	0.0070	2.842e-07
UF8	0.0510	0.0581	0.0629	0.0484	**0.0540**	0.0655	9.532e-05
UF9	0.0354	0.0399	0.1514	0.0369	0.1453	0.1497	0.2244
UF10	0.3531	0.4449	0.6221	0.3471	0.4500	0.6157	0.8087
R2_DTLZ2_M5	0.1093	0.1117	0.1143	0.1063	**0.1092**	0.1187	0.002342
R2_DTLZ3_M5	78.9882	140.9533	246.1105	64.3364	145.2939	194.5608	0.4853