Benchmark Functions for CEC'2018 Competition on Many-Objective Optimization

Ran Cheng[1], Miqing Li[1], Ye Tian[2], Xiaoshu Xiang[2],
Xingyi Zhang[2], Shengxiang Yang[3], Yaochu Jin[4], Xin Yao[1]

[1]CERCIA, School of Computer Science, University of Birmingham
Edgbaston, Birmingham B15 2TT, U.K.
{ranchengcn@gmail.com; limitsing@gmail.com; x.yao@cs.bham.ac.uk}
[2]School of Computer Science and Technology, Anhui University
Hefei 230039, China
{field910921@gmail.com; xyzhanghust@gmail.com; 1987385681@qq.com}
[3]School of Computer Science and Informatics, De Montfort University
Leicester, LE1 9BH, U.K.
{syang@dmu.ac.uk}
[4]Department of Computer Science, University of Surrey
Guildford, Surrey, GU2 7XH, U.K.
{yaochu.jin@surrey.ac.uk}

1 Introduction

The field of evolutionary multi-objective optimization has developed rapidly over the last two decades, but the design of effective algorithms for addressing problems with more than three objectives (called many-objective optimization problems, MaOPs) remains a great challenge. First, the ineffectiveness of the Pareto dominance relation, which is the most important criterion in multi-objective optimization, results in the underperformance of traditional Pareto-based algorithms. Also, the aggravation of the conflict between convergence and diversity, along with increasing time or space requirement as well as parameter sensitivity, has become key barriers to the design of effective many-objective optimization algorithms. Furthermore, the infeasibility of solutions' direct observation can lead to serious difficulties in algorithms' performance investigation and comparison. All of these suggest the pressing need of new methodologies designed for dealing with MaOPs, new performance metrics and benchmark functions tailored for experimental and comparative studies of evolutionary many-objective optimization (EMaO) algorithms.

In recent years, a number of new algorithms have been proposed for dealing with MaOPs [1], including the convergence enhancement based algorithms such as the grid-dominance based evolutionary algorithm (GrEA) [2], the knee point driven evolutionary algorithm (KnEA) [3], the two-archive algorithm (Two_Arch2) [4]; the decomposition based algorithms such as the NSGA-III [5], and the evolutionary algorithms based on both dominance and decomposition (MOEA/DD) [6], and the reference vector guided evolutionary algorithm (RVEA) [7]; and the performance indicator based algorithms such as the fast hypervolume based evolutionary algorithm (HypE) [8]. In spite of the various algorithms proposed for dealing with MaOPs, the literature still lacks a benchmark test suite for evolutionary many-objective optimization.

Benchmark functions play an important role in understanding the strengths and weaknesses of evolutionary algorithms. In many-objective optimization, several scalable continuous benchmark function suites, such as DTLZ [9] and WFG [10], have been commonly used. Recently,

researchers have also designed/presented some problem suites specially for many-objective optimization [11, 12, 13, 14, 15, 16]. However, all of these problem suites only represent one or several aspects of real-world scenarios. A set of benchmark functions with diverse properties for a systematic study of EMaO algorithms are not available in the area. On the other hand, existing benchmark functions typically have a "regular" Pareto front, overemphasize one specific property in a problem suite, or have some properties that appear rarely in real-world problems [17]. For example, the Pareto front of most of the DTLZ and WFG functions is similar to a simplex. This may be preferred by decomposition-based algorithms which often use a set of uniformly-distributed weight vectors in a simplex to guide the search [7, 18]. This simplex-like shape of Pareto front also causes an unusual property that any subset of all objectives of the problem can reach optimality [17, 19]. This property can be very problematic in the context of objective reduction, since the Pareto front degenerates into only one point when omitting one objective [19]. Also for the DTLZ and WFG functions, there is no function having a convex Pareto front; however, a convex Pareto front may bring more difficulty (than a concave Pareto front) for decomposition-based algorithms in terms of solutions' uniformity maintenance [20]. In addition, the DTLZ and WFG functions which are used as MaOPs with a degenerate Pareto front (i.e., DTLZ5, DTLZ6 and WFG3) have a nondegenerate part of the Pareto front when the number of objectives is larger than four [10, 21, 22]. This naturally affects the performance investigation of evolutionary algorithms on degenerate MaOPs.

This report slightly modifies the 15 test problems for many-objective optimization as originally given in [23]. The 15 benchmark problems are with diverse properties which cover a good representation of various real-world scenarios, such as being multimodal, disconnected, degenerate, and/or nonseparable, and having an irregular Pareto front shape, a complex Pareto set or a large number of decision variables (as summarized in Table 1). Our aim is to promote the research of many-objective optimization via suggesting a set of benchmark functions with a good representation of various real-world scenarios. As part of the competition, we provide the implementation of the test suites in two programming languages, i.e., MATLAB and Java. First, an open-source MATLAB software platform with a user-friendly GUI is provided inside the PlatEMO platform [24]. Second, a Java version is provided to be compatible with the jMetal platform [25]. In the following, Section 2 details the definitions of the 15 benchmark functions, and Section 3 presents the experimental setup for benchmark studies, including general settings, performance indicators, and software platforms.

Table 1: Main properties of the 15 test functions.

Problem	Properties	Note
MaF1	Linear	No single optimal solution in any subset of objectives
MaF2	Concave	No single optimal solution in any subset of objectives
MaF3	Convex, Multimodal	
MaF4	Concave, Multimodal	Badly-scaled and no single optimal solution in any subset of objectives
MaF5	Concave, Biased	Badly-scaled
MaF6	Concave, Degenerate	
MaF7	Mixed, Disconnected, Multimodal	
MaF8	Linear, Degenerate	
MaF9	Linear, Degenerate	Pareto optimal solutions are similar to their image in the objective space
MaF10	Mixed, Biased	
MaF11	Convex, Disconnected, Nonseparable	
MaF12	Concave, Nonseparable, Biased Deceptive	
MaF13	Concave, Unimodal, Nonseparable, Degenerate	Complex Pareto set
MaF14	Linear, Partially separable, Large scale	Non-uniform correlations between decision variables and objective functions
MaF15	Convex, Partially separable, Large scale	Non-uniform correlations between decision variables and objective functions

2 Function Definitions

- D: number of decision variables
- M: number of objectives
- $\mathbf{x} = (x_1, x_2, ..., x_D)$: decision vector
- f_i: i-th objective function

2.1 MaF1 (Modified inverted DTLZ1 [26])

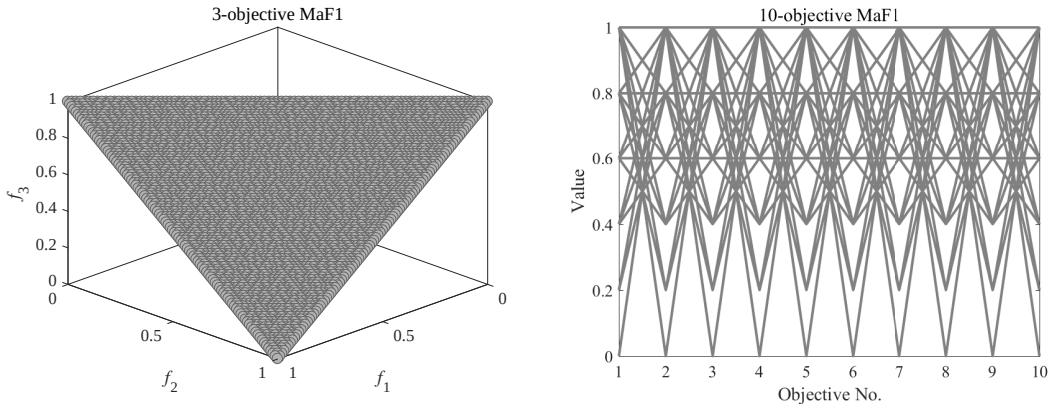

Figure 1: The Pareto front of MaF1 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = (1 - x_1 ... x_{M-1})(1 + g(\mathbf{x}_M)) \\ f_2(\mathbf{x}) = (1 - x_1 ... (1 - x_{M-1}))(1 + g(\mathbf{x}_M)) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = (1 - x_1(1 - x_2))(1 + g(\mathbf{x}_M)) \\ f_M(\mathbf{x}) = x_1(1 + g(\mathbf{x}_M)) \end{cases} \tag{1}$$

with

$$g(\mathbf{x}_M) = \sum_{i=M}^{|\mathbf{x}|} (x_i - 0.5)^2 \tag{2}$$

where the number of decision variable is $D = M + K - 1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. As shown in Fig. 1, this test problem has an inverted PF, while the PS is relatively simple. This test problem is used to assess whether EMaO algorithms are capable of dealing with inverted PFs. Parameter settings of this test problem are: $\mathbf{x} \in [0, 1]^D$ and $K = 10$.

2.2 MaF2 (DTLZ2BZ [19])

$$\min \begin{cases} f_1(\mathbf{x}) = \cos(\theta_1) ... \cos(\theta_2) \cos(\theta_{M-1})(1 + g_1(\mathbf{x}_M)) \\ f_2(\mathbf{x}) = \cos(\theta_1) ... \cos(\theta_{M-2}) \sin(\theta_{M-1})(1 + g_2(\mathbf{x}_M)) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = \cos(\theta_1) \sin(\theta_2)(1 + g_{M-1}(\mathbf{x}_M)) \\ f_M(\mathbf{x}) = \sin(\theta_1)(1 + g_M(\mathbf{x}_M)) \end{cases} \tag{3}$$

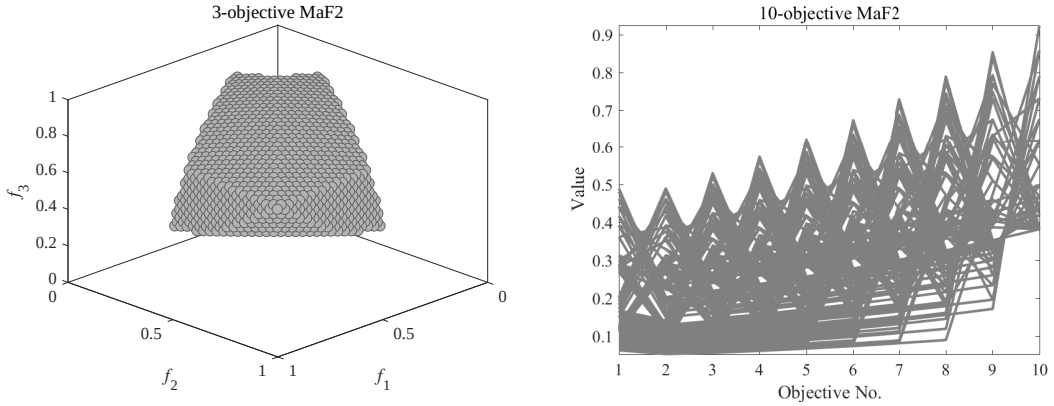

Figure 2: The Pareto front of MaF2 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

with

$$
\begin{aligned}
g_i(\mathbf{x}_M) &= \sum_{j=M+(i-1)\cdot\lfloor\frac{D-M+1}{M}\rfloor}^{M+i\cdot\lfloor\frac{D-M+1}{M}\rfloor-1} ((\frac{x_j}{2}+\frac{1}{4})-0.5)^2 \text{ for } i=1,\ldots,M-1 \\
g_M(\mathbf{x}_M) &= \sum_{j=M+(i-1)\cdot\lfloor\frac{D-M+1}{M}\rfloor}^{D} ((\frac{x_j}{2}+\frac{1}{4})-0.5)^2 \\
\theta_i &= \frac{\pi}{2}\cdot(\frac{x_i}{2}+\frac{1}{4}) \text{ for } i=1,\ldots,M-1
\end{aligned}
\tag{4}
$$

where the number of decision variable is $D = M+K-1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. This test problem is modified from DTLZ2 to increase the difficulty of convergence. In original DTLZ2, it is very likely that the convergence can be achieved once the $g(\mathbf{x}_M) = 0$ is satisfied; by contrast, for this modified version, all the objective have to be optimized simultaneously in order to reach the true PF. Therefore, this test problem is used to assess the whether and MOEA is able to perform concurrent convergence on different objectives. Parameter settings are: $\mathbf{x} \in [0,1]^D$ and $K = 10$.

2.3 MaF3 (Convex DTLZ3 [27])

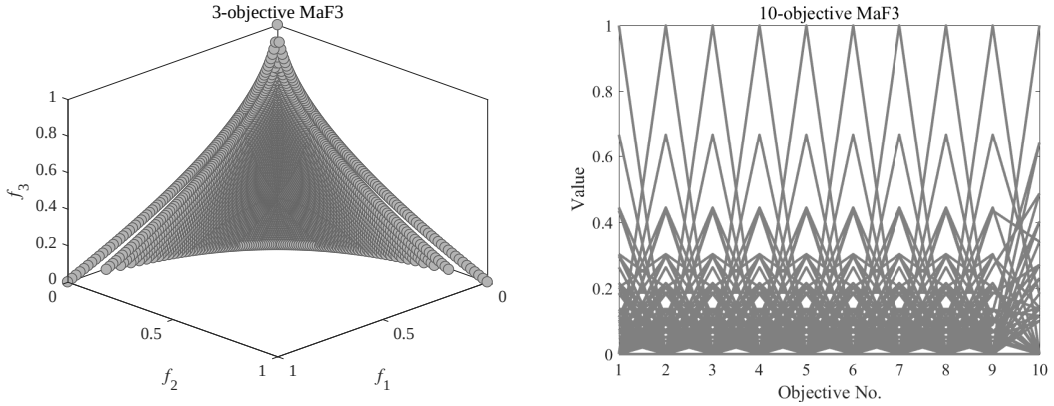

Figure 3: The Pareto front of MaF3 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = [\cos(\frac{\pi}{2}x_1)...\cos(\frac{\pi}{2}x_{M-2})\cos(\frac{\pi}{2}x_{M-1})(1+g(\mathbf{x}_M))]^4 \\ f_2(\mathbf{x}) = [\cos(\frac{\pi}{2}x_1)...\cos(\frac{\pi}{2}x_{M-2})\sin(\frac{\pi}{2}x_{M-1})(1+g(\mathbf{x}_M))]^4 \\ ... \\ f_{M-1}(\mathbf{x}) = [\cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1+g(\mathbf{x}_M))]^4 \\ f_M(\mathbf{x}) = [\sin(\frac{\pi}{2}x_1)(1+g(\mathbf{x}_M))]^2 \end{cases} \tag{5}$$

with

$$g(\mathbf{x}_M) = 100[|\mathbf{x}_M| + \sum_{i=M}^{|\mathbf{x}|}(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))] \tag{6}$$

where the number of decision variable is $D = M + K - 1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. As shown in Fig. 3, this test problem has a convex PF, and there a large number of local fronts. This test problem is mainly used to assess whether EMaO algorithms are capable of dealing with convex PFs. Parameter settings of this test problem are: $\mathbf{x} \in [0, 1]^D$, $K = 10$.

2.4 MaF4 (Inverted badly-scaled DTLZ3)

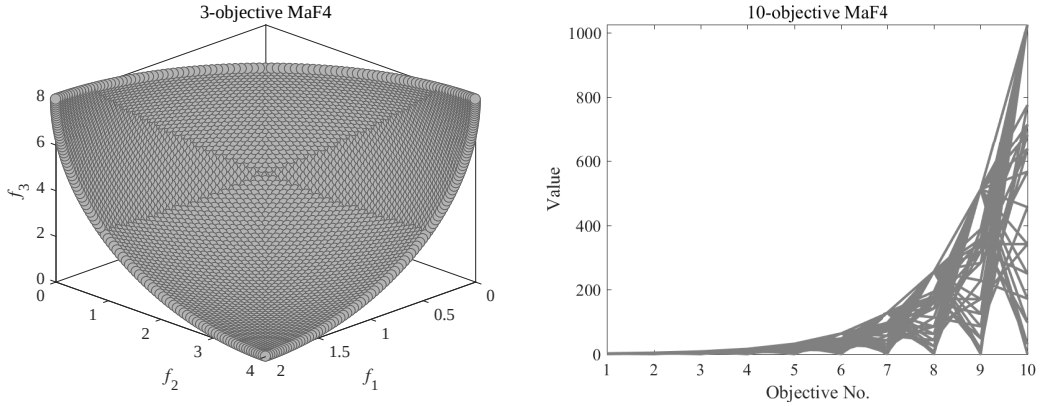

Figure 4: The Pareto front of MaF4 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = a \times (1 - \cos(\frac{\pi}{2}x_1)...\cos(\frac{\pi}{2}x_{M-2})\cos(\frac{\pi}{2}x_{M-1}))(1+g(\mathbf{x}_M)) \\ f_2(\mathbf{x}) = a^2 \times (1 - \cos(\frac{\pi}{2}x_1)...\cos(\frac{\pi}{2}x_{M-2})\sin(\frac{\pi}{2}x_{M-1}))(1+g(\mathbf{x}_M)) \\ ... \\ f_{M-1}(\mathbf{x}) = a^{M-1} \times (1 - \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2))(1+g(\mathbf{x}_M)) \\ f_M(\mathbf{x}) = a^M \times (1 - \sin(\frac{\pi}{2}x_1)) \times (1+g(\mathbf{x}_M)) \end{cases} \tag{7}$$

with

$$g(\mathbf{x}_M) = 100[|\mathbf{x}_M| + \sum_{i=M}^{|\mathbf{x}|}(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))] \tag{8}$$

where the number of decision variable is $D = M + K - 1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. Parameter settings are $a = 2$. Besides, the fitness landscape

of this test problem is highly multimodal, containing a number of $(3^k - 1)$ local Pareto-optimal fronts. This test problem is used to assess whether EMaO algorithms are capable of dealing with badly-scaled PFs, especially when the fitness landscape is highly multimodal. Parameter settings of this test problem are: $\mathbf{x} \in [0, 1]^D$, $K = 10$ and $a = 2$.

2.5 MaF5 (Concave badly-scaled DTLZ4)

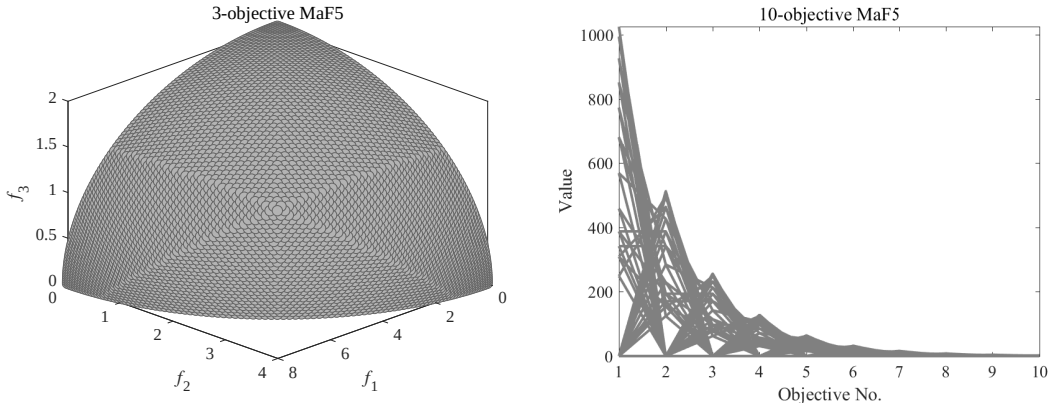

Figure 5: The Pareto front of MaF5 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = a^M \times [\cos(\frac{\pi}{2}{x_1}^\alpha)... \cos(\frac{\pi}{2}x_{M-2}^\alpha)\cos(\frac{\pi}{2}x_{M-1}^\alpha)(1 + g(\mathbf{x}_M))] \\ f_2(\mathbf{x}) = a^{M-1} \times [\cos(\frac{\pi}{2}{x_1}^\alpha)... \cos(\frac{\pi}{2}x_{M-2}^\alpha)\sin(\frac{\pi}{2}x_{M-1}^\alpha)(1 + g(\mathbf{x}_M))] \\ \quad ... \\ f_{M-1}(\mathbf{x}) = a^2 \times [\cos(\frac{\pi}{2}x_1^\alpha)\sin(\frac{\pi}{2}x_2^\alpha)(1 + g(\mathbf{x}_M))] \\ f_M(\mathbf{x}) = a \times [\sin(\frac{\pi}{2}{x_1}^\alpha)(1 + g(\mathbf{x}_M))] \end{cases} \tag{9}$$

with

$$g(\mathbf{x}_M) = \sum_{i=M}^{|\mathbf{x}|}(x_i - 0.5)^2 \tag{10}$$

where the number of decision variable is $D = M + K - 1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. As shown in Fig. 5, this test problem has a badly-scaled PF, where each objective function is scaled to a substantially different range. Besides, the PS of this test problem has a highly biased distribution, where the majority of Pareto optimal solutions are crowded in a small subregion. This test problem is used to assess whether EMaO algorithms are capable of dealing with badly-scaled PFs/PSs. Parameter settings of this test problem are: $\mathbf{x} \in [0, 1]^D$, $\alpha = 100$ and $a = 2$.

2.6 MaF6 (DTLZ5(I,M) [28])

$$\min \begin{cases} f_1(\mathbf{x}) = \cos(\theta_1)... \cos(\theta_{M-2})\cos(\theta_{M-1})(1 + 100g(\mathbf{x}_M)) \\ f_2(\mathbf{x}) = \cos(\theta_1)... \cos(\theta_{M-2})\sin(\theta_{M-1})(1 + 100g(\mathbf{x}_M)) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = \cos(\theta_1)\sin(\theta_2)(1 + 100g(\mathbf{x}_M)) \\ f_M(\mathbf{x}) = \sin(\theta_1)(1 + 100g(\mathbf{x}_M)) \end{cases} \tag{11}$$

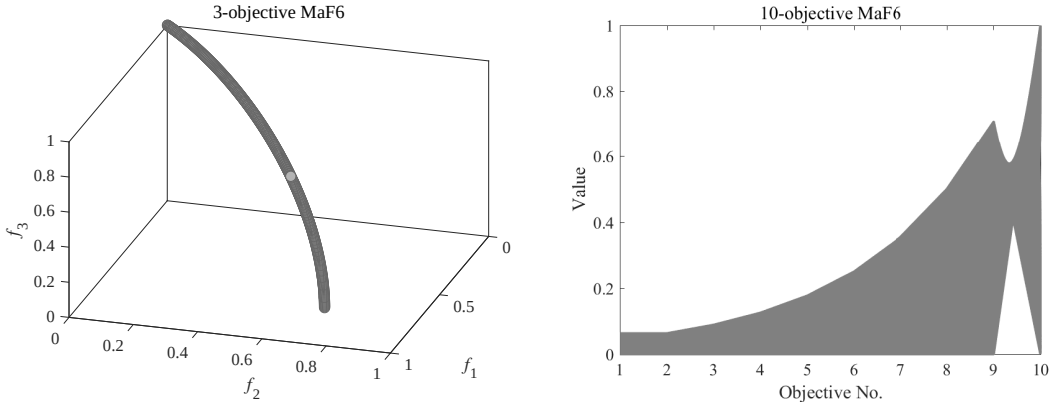

Figure 6: The Pareto front of MaF6 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

with

$$\theta_i = \begin{cases} \frac{\pi}{2} x_i \text{ for } i = 1, 2, ..., I-1 \\ \frac{\pi}{4(1+g(\mathbf{x}_M))}(1 + 2g(\mathbf{x}_M)x_i) \text{ for } i = I, ..., M-1 \end{cases} \tag{12}$$

$$g(\mathbf{x}_M) = \sum_{i=M}^{|\mathbf{x}|} (x_i - 0.5)^2 \tag{13}$$

where the number of decision variable is $D = M + K - 1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. As shown in Fig. 6, this test problem has a degenerate PF whose dimensionality is defined using parameter I. In other words, the PF of this test problem is always an I-dimensional manifold regardless of the specific number of decision variables. This test problem is used to assess whether EMaO algorithms are capable of dealing with degenerate PFs. Parameter settings are: $\mathbf{x} \in [0, 1]^D$, $I = 2$ and $K = 10$.

2.7 MaF7 (DTLZ7 [9])

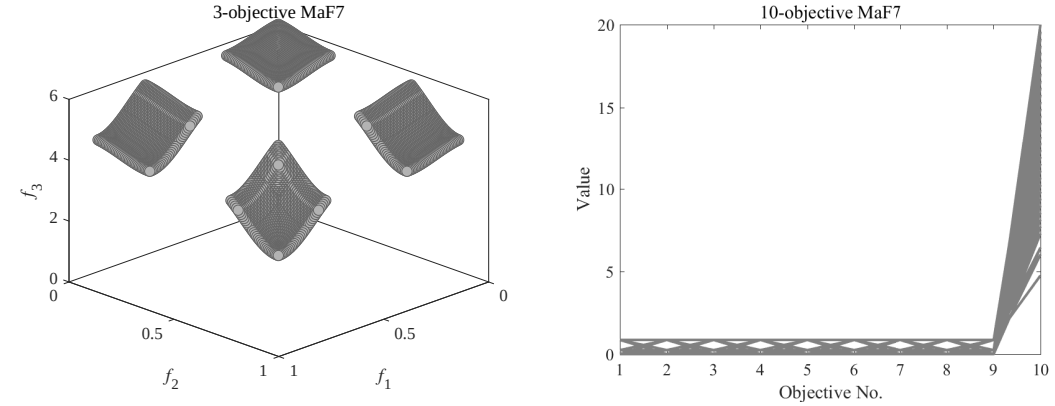

Figure 7: The Pareto front of MaF7 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = x_2 \\ \quad ... \\ f_{M-1}(\mathbf{x}) = x_{M-1} \\ f_M(\mathbf{x}) = h(f_1,\ f_2,\ \ldots,\ f_{M-1},\ g) \times (1 + g(\mathbf{x}_M)) \end{cases} \tag{14}$$

with

$$\begin{cases} g(\mathbf{x}_M) = 1 + \dfrac{9}{|\mathbf{x}_M|} \displaystyle\sum_{i=M}^{|\mathbf{x}|} x_i \\ h(f_1,\ f_2,\ \ldots,\ f_{M-1},\ g) = M - \displaystyle\sum_{i=1}^{M-1} [\frac{f_i}{1+g}(1 + \sin(3\pi f_i))] \end{cases} \tag{15}$$

where the number of decision variable is $D = M + K - 1$, and K denotes the size of $\mathbf{x}_M$, namely, $K = |\mathbf{x}_M|$, with $\mathbf{x}_M = (x_M, ..., x_D)$. As shown in Fig. 7, this test problem has a disconnected PF where the number of disconnected segments is 2^{M-1}. This test problem is used to assess whether EMaO algorithms are capable of dealing with disconnected PFs, especially when the number of disconnected segments is large in high-dimensional objective space. Parameter settings are: $\mathbf{x} \in [0, 1]^D$ and $K = 20$.

2.8 MaF8 (Multi-Point Distance Minimization Problem [11, 12])

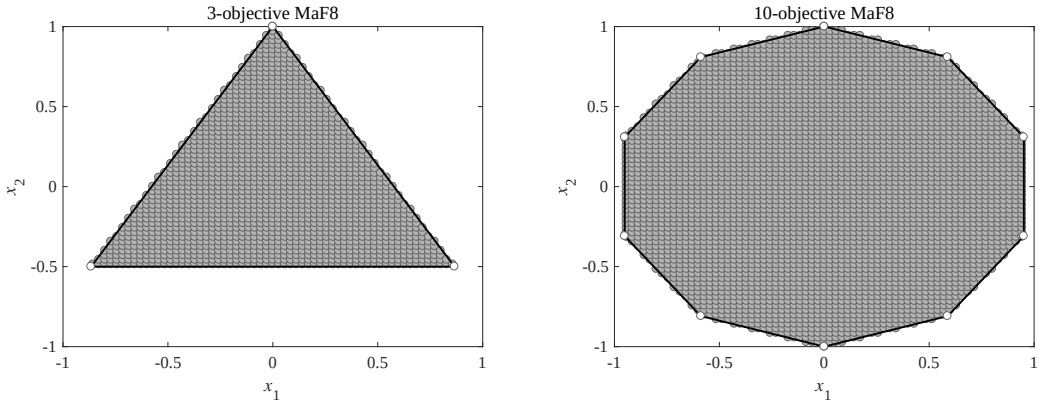

Figure 8: The Pareto front of MaF8 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

This function considers a two-dimensional decision space. As its name suggests, for any point $\mathbf{x} = (x_1, x_2)$ MaF8 calculates the Euclidean distance from $\mathbf{x}$ to a set of M target points $(A_1, A_2, ..., A_M)$ of a given polygon. The goal of the problem is to optimize these M distance values simultaneously. It can be formulated as

$$\min \begin{cases} f_1(\mathbf{x}) = d(\mathbf{x}, A_1) \\ f_2(\mathbf{x}) = d(\mathbf{x}, A_2) \\ \quad ... \\ f_M(\mathbf{x}) = d(\mathbf{x}, A_M) \end{cases} \tag{16}$$

where $d(\mathbf{x}, A_i)$ denotes the Euclidean distance from point $\mathbf{x}$ to point A_i.

One important characteristic of MaF8 is its Pareto optimal region in the decision space is typically a 2D manifold (regardless of the dimensionality of its objective vectors). This naturally

allows a direct observation of the search behavior of EMaO algorithms, e.g., the convergence of their population to the Pareto optimal solutions and the coverage of the population over the optimal region.

In this test suite, the regular polygon is used (in order to unify with MaF9). The center coordinates of the regular polygon (i.e., Pareto optimal region) are $(0, 0)$ and the radius of the polygon (i.e., the distance of the vertexes to the center) is 1.0. Parameter settings are: $\mathbf{x} \in [-10\ 000, 10\ 000]^2$. Fig. 8 shows the Pareto optimal regions of the 3-objective and 10-objective MaF8.

2.9 MaF9 (Multi-Line Distance Minimization Problem [29])

This function considers a two-dimensional decision space. For any point $\mathbf{x} = (x_1, x_2)$, MaF9 calculates the Euclidean distance from $\mathbf{x}$ to a set of M target straight lines, each of which passes through an edge of the given regular polygon with M vertexes $(A_1, A_2, ..., A_M)$, where $M \geq 3$. The goal of MaF9 is to optimize these M distance values simultaneously. It can be formulated as

$$\min \begin{cases} f_1(\mathbf{x}) = d(\mathbf{x}, \overleftrightarrow{A_1A_2}) \\ f_2(\mathbf{x}) = d(\mathbf{x}, \overleftrightarrow{A_2A_3}) \\ \quad ... \\ f_M(\mathbf{x}) = d(\mathbf{x}, \overleftrightarrow{A_MA_1}) \end{cases} \tag{17}$$

where $\overleftrightarrow{A_iA_j}$ is the target line passing through vertexes A_i and A_j of the regular polygon, and $d(\mathbf{x}, \overleftrightarrow{A_iA_j})$ denotes the Euclidean distance from point $\mathbf{x}$ to line $\overleftrightarrow{A_iA_j}$.

One key characteristic of MaF9 is that the points in the regular polygon (including the boundaries) and their objective images are similar in the sense of Euclidean geometry [29]. In other words, the ratio of the distance between any two points in the polygon to the distance between their corresponding objective vectors is a constant. This allows a straightforward understanding of the distribution of the objective vector set (e.g., its uniformity and coverage over the Pareto front) via observing the solution set in the two-dimensional decision space. In addition, for MaF9 with an even number of objectives ($M = 2k$ where $k \geq 2$), there exist k pairs of parallel target lines. Any point (outside the regular polygon) residing between a pair of parallel target lines is dominated by only a line segment parallel to these two lines. This property can pose a great challenge for EMaO algorithms which use Pareto dominance as the sole selection criterion in terms of convergence, typically leading to their populations trapped between these parallel lines [14].

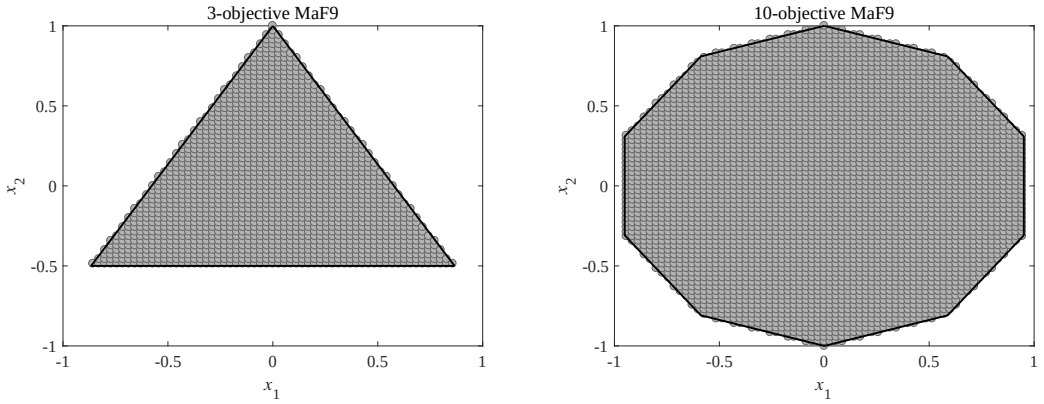

Figure 9: The Pareto front of MaF9 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

For MaF9, all points inside the polygon are the Pareto optimal solutions. However, these points may not be the sole Pareto optimal solutions of the problem. If two target lines intersect outside the regular polygon, there exist some areas whose points are nondominated with the interior points of the polygon. Apparently, such areas exist in the problem with five or more objectives in view of the convexity of the considered polygon. However, the geometric similarity holds only for the points inside the regular polygon. The Pareto optimal solutions that are located outside the polygon will affect this similarity property. So, we set some regions infeasible in the search space of the problem. Formally, consider an M-objective MaF9 with a regular polygon of vertexes $(A_1, A_2, ..., A_M)$. For any two target lines $\overleftrightarrow{A_{i-1}A_i}$ and $\overleftrightarrow{A_nA_{n+1}}$ (without loss of generality, assuming $i < n$) that intersect one point (O) outside the considered regular polygon, we can construct a polygon (denoted as $\Phi_{A_{i-1}A_iA_nA_{n+1}}$) bounded by a set of $2(n-i)+2$ line segments: $\overline{A_iA'_n}, \overline{A'_nA'_{n-1}}, ..., \overline{A'_{i+1}A'_i}, \overline{A'_iA_n}, \overline{A_nA_{n-1}}, ..., \overline{A_{i+1}A_i}$, where points $A'_i, A'_{i+1}, ..., A'_{n-1}, A'_n$ are symmetric points of $A_i, A_{i+1}, ...A_{n-1}, A_n$ with respect to central point O. We constrain the search space of the problem outside such polygons (but not including the boundary). Now the points inside the regular polygon are the sole Pareto optimal solutions of the problem. In the implementation of the test problem, for newly-produced individuals which are located in the constrained areas of the problem, we simply reproduce them within the given search space until they are feasible.

In this test suite, the center coordinates of the regular polygon (i.e., Pareto optimal region) are $(0, 0)$ and the radius of the polygon (i.e., the distance of the vertexes to the center) is 1.0. Parameter settings are: $\mathbf{x} \in [-10\,000, 10\,000]^2$. Fig. 9 shows the Pareto optimal regions of the 3-objective and 10-objective MaF9.

2.10 MaF10 (WFG1 [10])

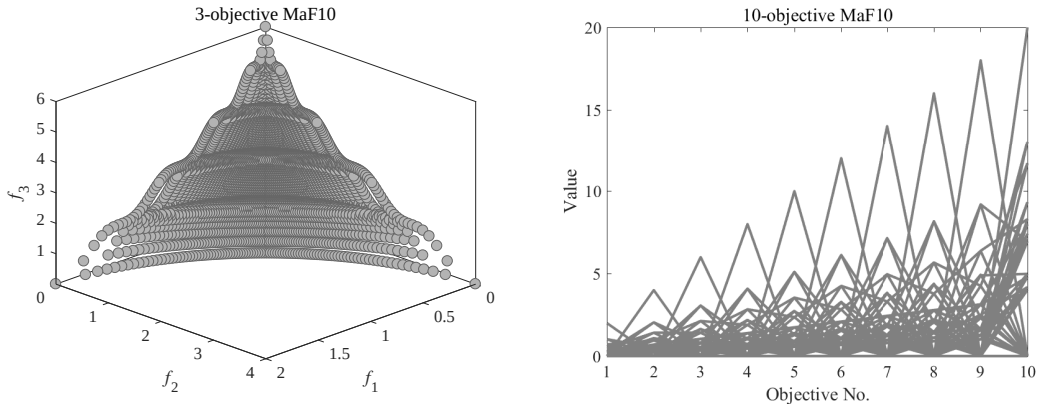

Figure 10: The Pareto front of MaF10 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = y_M + 2(1-\cos(\frac{\pi}{2}y_1))...(1-\cos(\frac{\pi}{2}y_{M-2}))(1-\cos(\frac{\pi}{2}y_{M-1})) \\ f_2(\mathbf{x}) = y_M + 4(1-\cos(\frac{\pi}{2}y_1))...(1-\cos(\frac{\pi}{2}y_{M-2}))(1-\sin(\frac{\pi}{2}y_{M-1})) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = y_M + 2(M-1)(1-\cos(\frac{\pi}{2}y_1))(1-\sin(\frac{\pi}{2}y_2)) \\ f_M(\mathbf{x}) = y_M + 2M(1-y_1-\frac{\cos(10\pi y_1+\pi/2)}{10\pi}) \end{cases} \tag{18}$$

with

$$z_i = \frac{x_i}{2i} \text{ for } i = 1, \ldots, D \tag{19}$$

$$t_i^1 = \begin{cases} z_i, & \text{if } i = 1, \ldots, K \\ \frac{|z_i - 0.35|}{|\lfloor 0.35 - z_i \rfloor + 0.35|}, & \text{if } i = K+1, \ldots, D \end{cases} \tag{20}$$

$$t_i^2 = \begin{cases} t_i^1, & \text{if } i = 1, \ldots, K \\ 0.8 + \frac{0.8(0.75 - t_i^1)\min(0, \lfloor t_i^1 - 0.75 \rfloor)}{0.75} - \frac{(1-0.8)(t_i^1 - 0.85)\min(0, \lfloor 0.85 - t_i^1 \rfloor)}{1-0.85}, & \text{if } i = K+1, \ldots, D \end{cases} \tag{21}$$

$$t_i^3 = t_i^{2^{0.02}} \text{ for } i = 1, \ldots, D \tag{22}$$

$$t_i^4 = \begin{cases} \frac{\sum_{j=(i-1)K/(M-1)+1}^{iK/(M-1)} 2j t_j^3}{\sum_{j=(i-1)K/(M-1)+1}^{iK/(M-1)} 2j}, & \text{if } i = 1, \ldots, M-1 \\ \frac{\sum_{j=K+1}^{D} 2j t_j^3}{\sum_{j=K+1}^{D} 2j}, & \text{if } i = M \end{cases} \tag{23}$$

$$y_i = \begin{cases} (t_i^4 - 0.5)\max(1, t_M^4) + 0.5, & \text{if } i = 1, \ldots, M-1 \\ t_M^4, & \text{if } i = M \end{cases} \tag{24}$$

where the number of decision variable is $D = K + L$, with K denoting the number of position variables and L denoting the number of distance variables. As shown in Fig. 10, this test problem has a scaled PF containing both convex and concave segments. Besides, there are a lot of transformation functions correlating the decision variables and the objective functions. This test problem is used to assess whether EMaO algorithms are capable of dealing with PFs of complicated mixed geometries. Parameter settings are: $\mathbf{x} \in \prod_{i=1}^{D}[0, 2i]$, $K = M - 1$, and $L = 10$.

2.11 MaF11 (WFG2 [10])

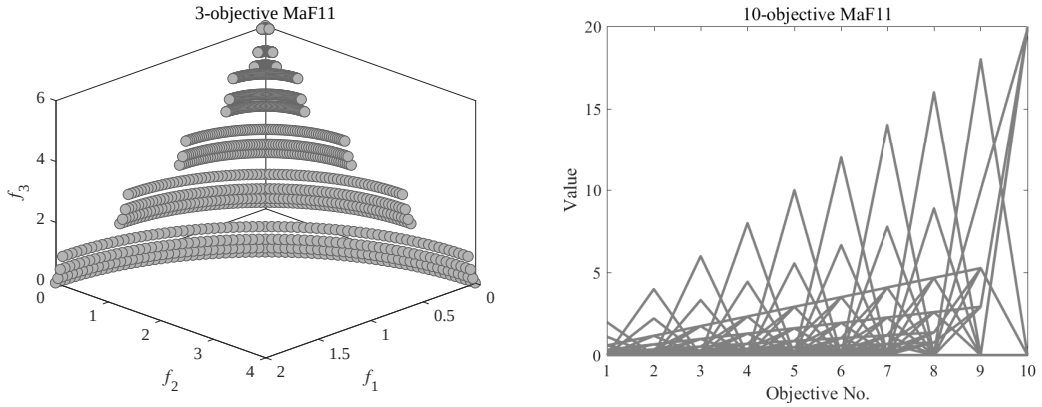

Figure 11: The Pareto front of MaF11 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = y_M + 2(1-\cos(\frac{\pi}{2}y_1))...(1-\cos(\frac{\pi}{2}y_{M-2}))(1-\cos(\frac{\pi}{2}y_{M-1})) \\ f_2(\mathbf{x}) = y_M + 4(1-\cos(\frac{\pi}{2}y_1))...(1-\cos(\frac{\pi}{2}y_{M-2}))(1-\sin(\frac{\pi}{2}y_{M-1})) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = y_M + 2(M-1)(1-\cos(\frac{\pi}{2}y_1))(1-\sin(\frac{\pi}{2}y_2)) \\ f_M(\mathbf{x}) = y_M + 2M(1-y_1\cos^2(5\pi y_1)) \end{cases} \tag{25}$$

with

$$z_i = \frac{x_i}{2i} \text{ for } i = 1, \ldots, D \tag{26}$$

$$t_i^1 = \begin{cases} z_i, & \text{if } i = 1, \ldots, K \\ \frac{|z_i - 0.35|}{|\lfloor 0.35 - z_i \rfloor + 0.35|}, & \text{if } i = K+1, \ldots, D \end{cases} \tag{27}$$

$$t_i^2 = \begin{cases} t_i^1, & \text{if } i = 1, \ldots, K \\ t_{K+2(i-K)-1}^1 + t_{K+2(i-K)}^1 + 2|t_{K+2(i-K)-1}^1 - t_{K+2(i-K)}^1|, & \text{if } i = K+1, \ldots, (D+K)/2 \end{cases} \tag{28}$$

$$t_i^3 = \begin{cases} \frac{\sum_{j=(i-1)K/(M-1)+1}^{iK/(M-1)} t_j^2}{K/(M-1)}, & \text{if } i = 1, \ldots, M-1 \\ \frac{\sum_{j=K+1}^{(D+K)/2} t_j^2}{(D-K)/2}, & \text{if } i = M \end{cases} \tag{29}$$

$$y_i = \begin{cases} (t_i^3 - 0.5)\max(1, t_M^3) + 0.5, & \text{if } i = 1, \ldots, M-1 \\ t_M^3, & \text{if } i = M \end{cases} \tag{30}$$

where the number of decision variable is $n = K + L$, with K denoting the number of position variables and L denoting the number of distance variables. As shown in Fig. 11, this test problem has a scaled disconnected PF. This test problem is used to assess whether EMaO algorithms are capable of dealing with scaled disconnected PFs. Parameter settings are: $\mathbf{x} \in \prod_{i=1}^{D}[0, 2i]$, $K = M - 1$, and $L = 10$.

2.12 MaF12 (WFG9 [10])

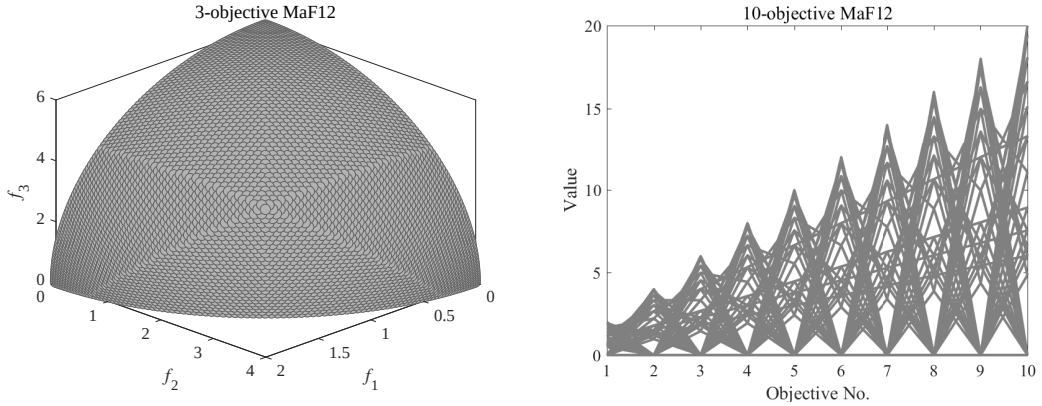

Figure 12: The Pareto front of MaF12 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = y_M + 2\sin(\frac{\pi}{2}y_1)...\sin(\frac{\pi}{2}y_{M-2})\sin(\frac{\pi}{2}y_{M-1}) \\ f_2(\mathbf{x}) = y_M + 4\sin(\frac{\pi}{2}y_1)...\sin(\frac{\pi}{2}y_{M-2})\cos(\frac{\pi}{2}y_{M-1}) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = y_M + 2(M-1)\sin(\frac{\pi}{2}y_1)\cos(\frac{\pi}{2}y_2) \\ f_M(\mathbf{x}) = y_M + 2M\cos(\frac{\pi}{2}y_1) \end{cases} \tag{31}$$

with

$$z_i = \frac{x_i}{2i} \text{ for } i = 1, \ldots, D \tag{32}$$

$$t_i^1 = \begin{cases} z_i^{0.02+(50-0.02)(0.98/49.98-(1-2\frac{\sum_{j=i+1}^{n} z_j}{D-i})|\lfloor 0.5-\frac{\sum_{j=i+1}^{D} z_j}{D-i}\rfloor+0.98/49.98|)}, & \text{if } i = 1, \ldots, D-1 \\ z_i, & \text{if } i = D \end{cases} \tag{33}$$

$$t_i^2 = \begin{cases} 1 + (|t_i^1 - 0.35| - 0.001)(\frac{349.95\lfloor t_i^1 - 0.349\rfloor}{0.349} + \frac{649.95\lfloor 0.351 - t_i^1\rfloor}{0.649} + 1000), & \text{if } i = 1, \ldots, K \\ \frac{1}{97}(1 + \cos[122\pi(0.5 - \frac{|t_i^1 - 0.35|}{2(\lfloor 0.35 - t_i^1\rfloor + 0.35)})] + 380(\frac{|t_i^1 - 0.35|}{2(\lfloor 0.35 - t_i^1\rfloor + 0.35)})^2), & \text{if } i = K+1, \ldots, D \end{cases} \tag{34}$$

$$t_i^3 = \begin{cases} \frac{\sum_{j=(i-1)K/(M-1)+1}^{iK/(M-1)} (t_j^2 + \sum_{k=0}^{K/(M-1)-2} |t_j^2 - t_p^2|)}{\lceil K/(M-1)/2\rceil(1+2K/(M-1)-2\lceil K/(M-1)/2\rceil)}, & \text{if } i = 1, \ldots, M-1 \\ \frac{\sum_{j=K+1}^{D} (t_j^2 + \sum_{k=0}^{D-K-2} |t_j^2 - t_q^2|)}{\lceil (D-K)/2\rceil(1+2(D-K)-2\lceil (D-K)/2\rceil)}, & \text{if } i = M \end{cases} \tag{35}$$

$$y_i = \begin{cases} (t_i^3 - 0.5)\max(1, t_M^3) + 0.5, & \text{if } i = 1, \ldots, M-1 \\ t_M^3, & \text{if } i = M \end{cases} \tag{36}$$

$$\begin{cases} p = (i-1)K/(M-1) + 1 + (j - (i-1)K/(M-1) + k)\text{mod}(K/(M-1)) \\ q = K + 1 + (j - K + k)\text{mod}(n - K) \end{cases} \tag{37}$$

where the number of decision variable is $D = K + L$, with K denoting the number of position variable and L denoting the number of distance variable. As shown in Fig. 12, this test problem has a scaled concave PF. Although the PF of this test problem is simple, its decision variables are nonseparably reduced, and its fitness landscape is highly multimodal. This test problem is used to assess whether EMaO algorithms are capable of dealing with scaled concave PFs together with complicated fitness landscapes. Parameter settings are: $\mathbf{x} \in \prod_{i=1}^{D}[0, 2i]$, $K = M - 1$, and $L = 10$.

2.13 MaF13 (PF7 [13])

$$\min \begin{cases} f_1(\mathbf{x}) = \sin(\frac{\pi}{2}x_1) + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2 \\ f_2(\mathbf{x}) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2) + \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2 \\ f_3(\mathbf{x}) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2) + \frac{2}{|J_3|}\sum_{j\in J_3} y_j^2 \\ f_{4,\ldots,M}(\mathbf{x}) = f_1(\mathbf{x})^2 + f_2(\mathbf{x})^{10} + f_3(\mathbf{x})^{10} + \frac{2}{|J_4|}\sum_{j\in J_4} y_j^2 \end{cases} \tag{38}$$

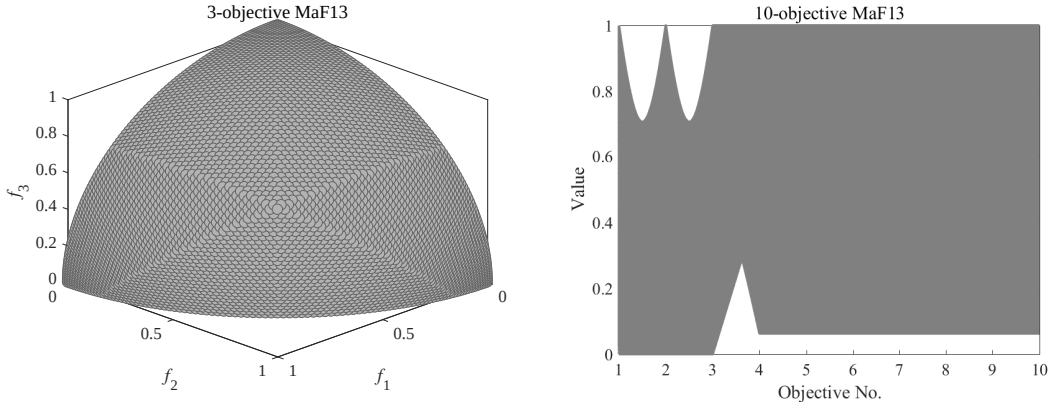

Figure 13: The Pareto front of MaF13 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

with

$$y_i = x_i - 2x_2 \sin(2\pi x_1 + \frac{i\pi}{n}) \text{ for } i = 1, \ldots, D \tag{39}$$

$$\begin{cases} J_1 = \{j | 3 \leq j \leq D, \text{ and } j \bmod 3 = 1\} \\ J_2 = \{j | 3 \leq j \leq D, \text{ and } j \bmod 3 = 2\} \\ J_3 = \{j | 3 \leq j \leq D, \text{ and } j \bmod 3 = 0\} \\ J_4 = \{j | 4 \leq j \leq D\} \end{cases} \tag{40}$$

where the number of decision variable is $D = 5$. As shown in Fig. 13, this test problem has a concave PF; in fact, the PF of this problem is always a unit sphere regardless of the number of objectives. Although this test problem has a simple PF, its decision variables are nonlinearly linked with the first and second decision variables, thus leading to difficulty in convergence. This test problem is used to assess whether EMaO algorithms are capable of dealing with degenerate PFs and complicated variable linkages. Parameter setting is: $\mathbf{x} \in [0, 1]^2 \times [-2, 2]^{D-2}$.

2.14 MaF14 (LSMOP3 [16])

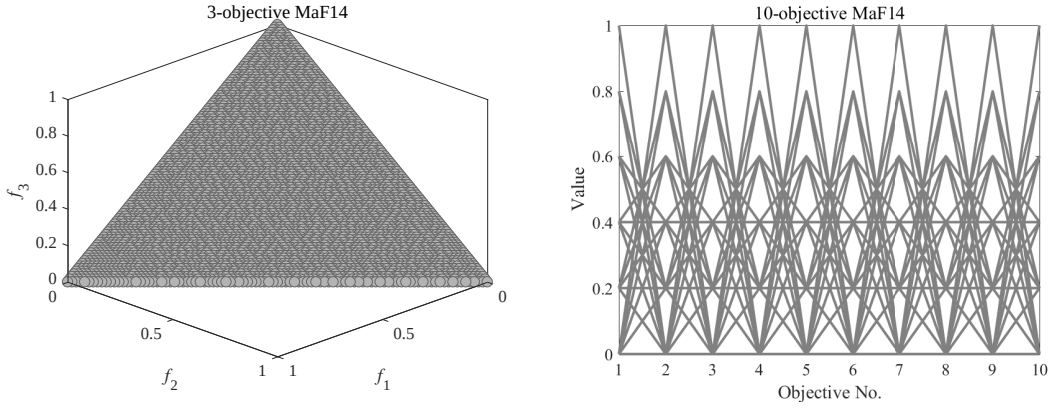

Figure 14: The Pareto front of MaF14 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

$$\min \begin{cases} f_1(\mathbf{x}) = x_1^f ... x_{M-1}^f (1 + \sum_{j=1}^{M} c_{1,j} \times \bar{g}_1(\mathbf{x}_j^s)) \\ f_2(\mathbf{x}) = x_1^f ... (1 - x_{M-1}^f)(1 + \sum_{j=1}^{M} c_{2,j} \times \bar{g}_2(\mathbf{x}_j^s)) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = x_1^f (1 - x_2^f)(1 + \sum_{j=1}^{M} c_{M-1,j} \times \bar{g}_{M-1}(\mathbf{x}_j^s)) \\ f_M(\mathbf{x}) = (1 - x_1^f)(1 + \sum_{j=1}^{M} c_{M,j} \times \bar{g}_M(\mathbf{x}_j^s)) \\ \mathbf{x} \in [0, 10]^{|\mathbf{x}|} \end{cases} \tag{41}$$

with

$$c_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

$$\begin{cases} \bar{g}_{2k-1}(\mathbf{x}_i^s) = \dfrac{1}{N_k} \sum_{j=1}^{N_k} \dfrac{\eta_1(\mathbf{x}_{i,j}^s)}{|\mathbf{x}_{i,j}^s|} \\ \bar{g}_{2k}(\mathbf{x}_i^s) = \dfrac{1}{N_k} \sum_{j=1}^{N_k} \dfrac{\eta_2(\mathbf{x}_{i,j}^s)}{|\mathbf{x}_{i,j}^s|} \\ k = 1, ..., \lceil \dfrac{M}{2} \rceil \end{cases} \tag{43}$$

$$\begin{cases} \eta_1(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} (x_i^2 - 10\cos(2\pi x_i) + 10) \\ \eta_2(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \end{cases} \tag{44}$$

$$\begin{cases} \mathbf{x}^s \leftarrow (1 + \dfrac{i}{|\mathbf{x}^s|}) \times (x_i^s - l_i) - x_1^f \times (u_i - l_i) \\ i = 1, ..., |\mathbf{x}^s| \end{cases} \tag{45}$$

where N_k denotes the number of variable subcomponent in each variable group $\mathbf{x}_i^s$ with $i = 1, ..., M$, and u_i and l_i are the upper and lower boundaries of the i-th decision variable in $\mathbf{x}^s$. Although this test problem has a simple linear PF, its fitness landscape is complicated. First, the decision variables are non-uniformly correlated with different objectives; second, the decision variables have mixed separability, i.e., some of the are separable while others are not. This test problem is mainly used to assess whether EMaO algorithms are capable of dealing with complicated fitness landscape with mixed variable separability, especially in large-scale cases. Parameter settings are: $N_k = 2$ and $D = 20 \times M$.

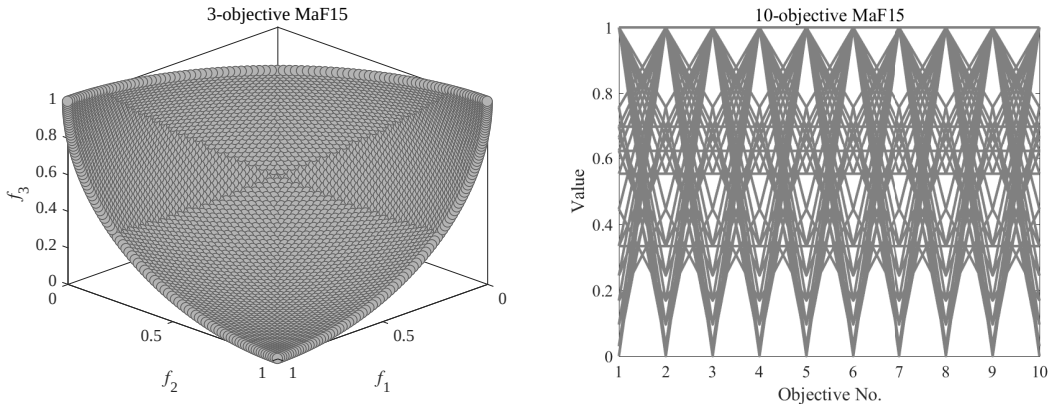

Figure 15: The Pareto front of MaF15 with 3 and 10 objectives, shown by Cartesian coordinates and parallel coordinates respectively.

2.15 MaF15 (Inverted LSMOP8 [16])

$$\min \begin{cases} f_1(\mathbf{x}) = (1-\cos(\frac{\pi}{2}x_1^f)...\cos(\frac{\pi}{2}x_{M-2}^f)\cos(\frac{\pi}{2}x_{M-1}^f)) \times (1+\sum_{j=1}^{M} c_{1,j} \times \bar{g}_1(\mathbf{x}_j^s)) \\ f_2(\mathbf{x}) = (1-\cos(\frac{\pi}{2}x_1^f)...\cos(\frac{\pi}{2}x_{M-2}^f)\sin(\frac{\pi}{2}x_{M-1}^f)) \times (1+\sum_{j=1}^{M} c_{2,j} \times \bar{g}_2(\mathbf{x}_j^s)) \\ \quad ... \\ f_{M-1}(\mathbf{x}) = (1-\cos(\frac{\pi}{2}x_1^f)\sin(\frac{\pi}{2}x_2^f)) \times (1+\sum_{j=1}^{M} c_{M-1,j} \times \bar{g}_{M-1}(\mathbf{x}_j^s)) \\ f_M(\mathbf{x}) = (1-\sin(\frac{\pi}{2}x_1^f)) \times (1+\sum_{j=1}^{M} c_{M,j}\bar{g}_M(\mathbf{x}_j^s)) \\ \mathbf{x} \in [0,1]^{|\mathbf{x}|} \end{cases} \tag{46}$$

with

$$c_{i,j} = \begin{cases} 1, & \text{if } j = i \text{ or } j = i+1 \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

$$\begin{cases} \bar{g}_{2k-1}(\mathbf{x}_i^s) = \dfrac{1}{N_k}\sum_{j=1}^{N_k} \dfrac{\eta_1(\mathbf{x}_{i,j}^s)}{|\mathbf{x}_{i,j}^s|} \\ \bar{g}_{2k}(\mathbf{x}_i^s) = \dfrac{1}{N_k}\sum_{j=1}^{N_k} \dfrac{\eta_2(\mathbf{x}_{i,j}^s)}{|\mathbf{x}_{i,j}^s|} \\ k = 1, ..., \lceil \dfrac{M}{2} \rceil \end{cases} \tag{48}$$

$$\begin{cases} \eta_1(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} \dfrac{x_i^2}{4000} - \prod_{i=1}^{|\mathbf{x}|} \cos(\dfrac{x_i}{\sqrt{i}}) + 1 \\ \eta_2(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} (x_i)^2\}. \end{cases} \tag{49}$$

$$\begin{cases} \mathbf{x}^s \leftarrow (1+\cos(0.5\pi\frac{i}{|\mathbf{x}^s|}))\times(x_i^s-l_i)-x_1^f\times(u_i-l_i) \\ i=1,...,|\mathbf{x}^s| \end{cases} \tag{50}$$

where N_k denotes the number of variable subcomponent in each variable group $\mathbf{x}_i^s$ with $i = 1, ..., M$, and u_i and l_i are the upper and lower boundaries of the i-th decision variable in $\mathbf{x}^s$. Although this test problem has a simple convex PF, its fitness landscape is complicated. First, the decision variables are non-uniformly correlated with different objectives; second, the decision variables have mixed separability, i.e., some of the are separable while others are not. Different from MaF14, this test problem has non-linear (instead of linear) variable linkages on the PS, which further increases the difficulty. This test problem is mainly used to assess whether EMaO algorithms are capable of dealing with complicated fitness landscape with mixed variable separability, especially in large-scale cases. Parameter settings are: $N_k = 2$ and $D = 20 \times M$.

3 Experimental Setup

To conduct benchmark experiments using the proposed test suite, users may follow the experimental setup as given below.

3.1 General Settings

- **Number of Objectives (M)**: 5, 10, 15
- **Maximum Population Size**[1]: 240
- **Maximum Number of Fitness Evaluations (FEs)**[2]: $\max\{100000, 10000 \times D\}$
- **Number of Independent Runs**: 20

3.2 Performance Metrics

- **Inverted Generational Distance (IGD)**: Let P^* be a set of uniformly distributed points on the Pareto front. Let P be an approximation to the Pareto front. The inverted generational distance between P^* and P can be defined as:

$$IGD(P^*, P) = \frac{\sum_{\mathbf{v}\in P^*} d(v, P)}{|P^*|}, \tag{51}$$

  where $d(\mathbf{v}, P)$ is the minimum Euclidean distance from point $\mathbf{v}$ to set P. The IGD metric is able to measure both diversity and convergence of P if $|P^*|$ is large enough, and a smaller IGD value indicates a better performance. In this test suite, we suggest a number of 10000 uniformly distributed reference points sampled on the true Pareto front[3] for each test instance.

- **Hypervolume (HV)**: Let $\mathbf{y}^* = (y_1^*, ..., y_m^*)$ be a reference point in the objective space that is dominated by all Pareto optimal solutions. Let P be the approximation to the Pareto front. The HV value of P (with regard to $\mathbf{y}^*$) is the volume of the region which is dominated by P and dominates $\mathbf{y}^*$. In this test suite, the objective vectors in P are

[1]The size of final population/archive must be smaller the given maximum population size, otherwise, a compulsory truncation will be operated in final statistics for fair comparisons.

[2]Regardless of the number of objectives, every evaluation of the whole objective set is counted as one FE.

[3]The specific number of reference points for IGD calculations can vary a bit due to the different geometries of the Pareto fronts. All reference point sets can be automatically generated using the software platform introduced in Section 3.3.

normalized using $f_i^j = \frac{f_i^j}{1.1 \times y_i^{nadir}}$, where f_i^j is the i-th dimension of j-th objective vector, and y_i^{nadir} is the i-th dimension of nadir point of the true Pareto front[4]. Then we use y* = (1,...,1) as the reference point for the normalized objective vectors in the HV calculation.

3.3 Matlab Software Platform

All the benchmark functions have been implemented in MATLAB code and embedded in a recently developed software platform – PlatEMO[5]. PlatEMO is an open source MATLAB-based platform for evolutionary multi- and many-objective optimization, which currently includes more than 50 representative algorithms and more than 100 benchmark functions, along with a variety of widely used performance indicators. Moreover, PlatEMO provides a user-friendly graphical user interface (GUI), which enables users to easily perform experimental settings and algorithmic configurations, and obtain statistical experimental results by one-click operation.

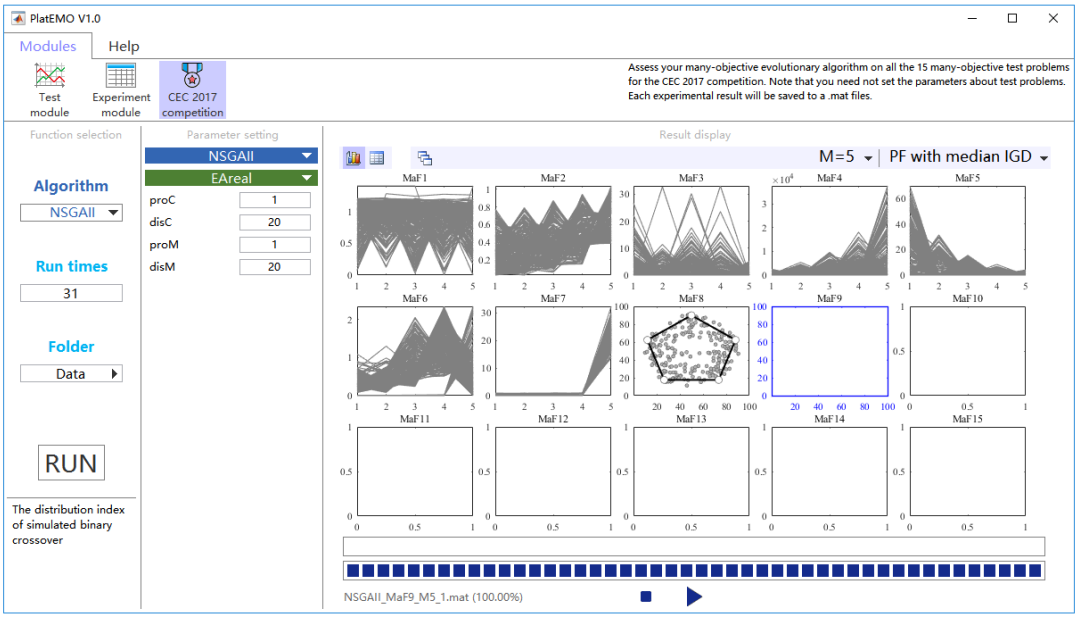

Figure 16: The GUI in PlatEMO for this test suite.

In particular, as shown in Fig. 16, we have tailored a new GUI in PlatEMO for this test suite, such that participants are able to directly obtain tables and figures comprising the statistical experimental results for the test suite. To conduct the experiments, the only thing to be done by participants is to write the candidate algorithms in MATLAB and embed them into PlatEMO. The detailed introduction to PlatEMO regarding how to embed new algorithms can be referred to the users manual attached in the source code of PlatEMO [24]. Once a new algorithm is embedded in PlatEMO, the user will be able to select the new algorithm and execute it on the GUI shown in Fig. 16. Then the statistical results will be displayed in the figures and tables on the GUI, and the corresponding experimental result (i.e. final population and its performance indicator values) of each run will be saved to a *.mat* file.

3.4 Java Software Platform

The benchmark functions implemented in Java code have been embedded in an object-oriented framework – jMetal[6]. jMetal, which stands for Metaheuristic Algorithms in Java, is a Java-

[4]The nadir points can be automatically generated using the software platform introduced in Section 3.3.
[5]PlatEMO can be downloaded at `http://bimk.ahu.edu.cn/index.php?s=/Index/Software/index.html`
[6]jMetal can be downloaded at `http://jmetal.sourceforge.net/index.html`

based framework for multi-objective optimization with metaheuristics developed in recent years by J.J. Durillo [25]. The framework provides more than 30 classic and state-of-the-art algorithms, including single or multi-objective evolutionary algorithms as well as other metaheuristics. Nearly 70 benchmark problems and a set of well-known quality indicators are included to assess the performance of these algorithms. Furthermore, full experimental studies can be configured and executed with a graphical interface supported.

With the experimental support GUI, users can select algorithms, problems and indicators and configure the experiment properly. The final results can be stored and generated as Latex tables, Wilcoxon tests or boxplots. For carrying out newly developed algorithms or other settings, the modification of code based on jMetal must be made from the user's point of view. By embedding these algorithms into jMetal, the user can execute them with code or GUI in a similar way to other existing algorithms.

References

[1] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, p. 13, 2015.

[2] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[3] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[4] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, Aug 2015.

[5] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[6] K. Li, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[7] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[8] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds. Berlin, Germany: Springer, 2005, pp. 105–145.

[10] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[11] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization (EMO)*, 2007, pp. 727–741.

[12] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," in *International Conference on Parallel Problem Solving from Nature (PPSN)*, 2010, pp. 91–100.

[13] D. Saxena, Q. Zhang, J. Duro, and A. Tiwari, "Framework for many-objectivet test problems with both simple and complicated Pareto-set shapes," in *Evolutionary Multi-Criterion Optimization (EMO)*, 2011, pp. 197–211.

[14] M. Li, S. Yang, and X. Liu, "A test problem for visual investigation of high-dimensional multi-objective search," in *IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 2140–2147.

[15] Y.-M. Cheung, F. Gu, and H.-L. Liu, "Objective extraction for many-objective optimization problems: Algorithm and test problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 755–772, 2016.

[16] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "Test problems for large-scale multiobjective and many-objective optimization." *IEEE Transactions on Cybernetics*, 2016, in press.

[17] H. Masuda, Y. Nojima, and H. Ishibuchi, "Common properties of scalable multiobjective problems and a new framework of test problems," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 3011–3018.

[18] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 127–140.

[19] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: Theory and applications," *Evolutionary Computation*, vol. 17, no. 2, pp. 135–166, 2009.

[20] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2016.

[21] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[22] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 807–813, 2016.

[23] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

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

[25] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, pp. 760–771, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997811001219

[26] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, Aug 2014.

[27] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[28] K. Deb and D. K. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *IEEE Congress on Evolutionary Computation (CEC)*, 2006, pp. 3353–3360.

[29] M. Li, C. Grosan, S. Yang, X. Liu, and X. Yao, "Multi-line distance minimization: A visualized many-objective test problem suite," *IEEE Transactions on Evolutionary Computation*, 2017, in press.