Preference-Inspired Coevolutionary Algorithms for Many-Objective Optimization

Rui Wang, Robin C. Purshouse, and Peter J. Fleming

***Abstract*—The simultaneous optimization of many objectives (in excess of 3), in order to obtain a full and satisfactory set of tradeoff solutions to support *a posteriori* decision making, remains a challenging problem. The concept of coevolving a family of decision-maker preferences together with a population of candidate solutions is studied here and demonstrated to have promising performance characteristics for such problems. After introducing the concept of the preference-inspired coevolutionary algorithm (PICEA), a realization of this concept, PICEA-g, is systematically compared with four of the best-in-class evolutionary algorithms (EAs); random search is also studied as a baseline approach. The four EAs used in the comparison are a Pareto-dominance relation-based algorithm (NSGA-II), an ϵ-dominance relation-based algorithm [ϵ-multiobjective evolutionary algorithm (MOEA)], a scalarizing function-based algorithm (MOEA/D), and an indicator-based algorithm [hypervolume-based algorithm (HypE)]. It is demonstrated that, for bi-objective problems, all of the multi-objective evolutionary algorithms perform competitively. As the number of objectives increases, PICEA-g and HypE, which have comparable performance, tend to outperform NSGA-II, ϵ-MOEA, and MOEA/D. All the algorithms outperformed random search.**

***Index Terms*—Coevolution, evolutionary algorithms, many-objective optimization.**

I. Introduction

IT IS an accepted fact that multiobjective evolutionary algorithms (MOEAs) can be successfully applied to multiobjective optimization problems (MOPs) possessing two or three objectives [1], [2]. However, more recent studies have suggested that the search ability of some MOEAs, e.g., NSGA-II [3] and SPEA2 [4], is often severely degraded by an increase in the number of objectives [5], [6]. Particularly, the "sweet-spot" [7] of algorithm parameter settings that yield good performance may greatly contract [5], [6], i.e., algorithms become more sensitive to the user's choice of parameter settings.

Manuscript received September 5, 2011; revised February 23, 2012 and June 1, 2012; accepted June 4, 2012. Date of publication June 11, 2012; date of current version July 25, 2013. The work of R. Wang was supported by the China Scholarship Council and the National Science Foundation of China, under Grant 70971132.

R. Wang is with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K., and also with the Department of Systems Engineering, College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China (e-mail: cop10rw@sheffield.ac.uk).

R. C. Purshouse and P. J. Fleming are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: r.purshouse@sheffield.ac.uk; p.fleming@shef.ac.uk).

Digital Object Identifier 10.1109/TEVC.2012.2204264

MOPs with more than three objectives are significantly more challenging; these have been termed many-objective optimization problems [6], [8]. The poor performance of Pareto-dominance-based MOEAs is due to the fact that the proportion of nondominated objective vectors in each MOEA population becomes very large as the number of objectives increases. As a result, not enough selective pressure can be generated toward the Pareto front [6], [8]–[10].

Various approaches have been proposed to improve the search ability of standard Pareto-dominance-based MOEAs for many-objective problems [10], [11]. In [11], five different scalability improvement approaches are considered: 1) modification of the density estimator [12], [13]; 2) introduction of different ranks [13]–[15]; 3) modification of Pareto-dominance [16], [17]; 4) modification of objective functions [17], [18]; and 5) hybridization with local search [19]. Ishibuchi *et al.* examined the effects of these approaches on the performance of NSGA-II, demonstrating empirically that most of these approaches can improve the convergence property but simultaneously decrease the diversity of obtained solutions [11]. Aside from these approaches, a relaxed form of Pareto dominance known as ϵ-dominance was proposed by Laumanns *et al.* as a means of maintaining both the convergence and diversity of solutions [20]. Wagner *et al.* demonstrated that ϵ-dominance-based MOEAs are promising approaches for many-objective problems [12].

A theoretically well-supported alternative to Pareto dominance is the use of an indicator function to measure the quality of solution sets. This kind of MOEA is referred to as an indicator-based evolutionary algorithm (IBEA) [21]. The hypervolume indicator [22], which is the only unary quality measure that is strictly monotonic with regard to Pareto dominance, is often used as an indicator [23], [24]. However, the high computational effort required for hypervolume calculation [25] greatly inhibits its application. Recently, Bader and Zitzler proposed a new hypervolume-based algorithm (HypE), in which a Monte Carlo simulation method is used to approximate the exact hypervolume value [26], [27]. This approximation method significantly reduces computation load and makes HypE very competitive for solving many-objective optimization problems.

The scalarizing function-based fitness evaluation approach (i.e., weighted sum or weighted Tchebycheff) is another promising alternative to the Pareto-dominance relation. The most well-known representative MOEA based on this concept is scalarizing function-based algorithm (MOEA/D) [28]. It has been demonstrated in the literature [9], [28]–[30] that MOEA/D has high search ability for continuous optimization, combinatorial optimization, and also performs well on problems with complex Pareto sets. MOEA/D, the winner of the Unconstrained Multiobjective Evolutionary Algorithm Competition at the 2009 Congress on Evolutionary Computation [29], is an important approach to consider for solving many-objective optimization problems.

Since preference-based approaches are useful for the generation of tradeoff surfaces in objective subspaces of interest to the decision maker [31], [32], intuitively, coevolving a family of preferences simultaneously with the population of candidate solutions has the potential to be another promising concept for solving many-objective problems. We refer to realizations of this concept as preference-inspired coevolutionary algorithms (PICEAs), since the preferences are being used to generate approximation sets for *a posteriori* decision making, rather than representing true articulations of decision-maker preferences for *a priori* or progressive optimization. Recent research by Purshouse *et al.* [33] demonstrated that an approach of this type is able to outperform NSGA-II, the average ranking method [14], and random search on many-objective problems. In this paper, we further explore the potential of the PICEA concept by comparing a variant of the Purshouse *et al.* [33] algorithm with representative algorithms from other classes of MOEA: the ϵ-dominance-based algorithm, ϵ-MOEA [34], an indicator-based algorithm, HypE [27], a MOEA/D [28], and the Pareto-dominance-based algorithm, NSGA-II [3], along with random search that is included as a baseline.

The remainder of this paper is structured as follows. Section II first introduces the concept of a PICEA and provides one realization based on a particular definition of preference. This is followed, in Section III, by brief descriptions of the representative search methods used for comparison purposes. Section IV introduces the proposed test functions, performance measures, and general parameter settings to be used in the experiments undertaken for the comparison study. Experimental results are presented in Section V. Section VI offers a further discussion of the algorithms. Findings, limitations of the study, and proposals for future research are presented in Section VII. Finally, conclusions are drawn in Section VIII.

II. Preference-Inspired Coevolutionary Algorithms

A. Motivation

One of the major challenges identified for many-objective optimization is the reduced ability of the Pareto-dominance relation in offering comparability between alternative solutions [2]. This lack of comparability means that algorithms using Pareto-dominance struggle to drive the search toward the Pareto front [6]. However, it has long been known that by using decision-maker preferences, we can potentially gain comparability between otherwise incomparable solutions [35]. A classic example can be found in the early seminal work of Fonseca and Fleming [31], in which the effect of different specifications of decision-maker goals and priorities on the partial ordering of solutions in an enumerated search space is shown with striking clarity. If we specify a single set of preferences, then we focus the search on a subset of the Pareto front (possibly a single solution, depending on how the preferences are specified), with different sets of preferences leading to different subsets of the front.

However, our interest here remains in *a posteriori* optimization, i.e., providing decision makers with both a proximal and a diverse representation of the entire Pareto front [36], prior to the elicitation and application of their preferences. In this sense, we are interested in holding multiple sets of hypothetical preferences simultaneously, to provide multiple comparison perspectives simultaneously, which are sufficient to adequately describe the whole front. The simultaneity is what differentiates this approach from the multiple restart strategies of conventional multicriteria decision-making methods [35].

The notion of using preferences in this way (i.e., not as real decision-maker preferences but as a means of comparing solutions for the purposes of *a posteriori* optimization) is not new to evolutionary multi-criteria optimization (EMO) research, but is certainly underexplored. Existing approaches have tended to focus on an aggregation-based formulation of preferences; Jin *et al.* considered preferences in the form of a weighted sum, in which the weightings were varied over the course of the search [37]. Hughes proposed a ranking method based on how each solution performed against a set of predefined preferences expressed as weighted min–max formulations [38]. Zhang *et al.* developed a cellular algorithm in which each node represents a particular preference formulation, leading to a spatial distribution of preferences [28]. The challenge for these methods is how to define a suitable family of preferences that will produce a full representation of the Pareto front. This issue is slightly different from that faced by multiple restart algorithms, since the focus is on maintaining the usefulness of the preference family during the course of a population-based search. Few approaches of this type are known. Hughes extended his earlier work to consider online generation of target vectors by bootstrapping these from the online archive of locally nondominated solutions, with mixed results [39].

A potential way of maintaining the relevance of the preference family as the search progresses is to coevolve the family together with the usual population of candidate solutions. The solutions would gain fitness by performing well against the preferences (as in the approaches above), and the preferences would gain fitness by offering comparability between solutions. In our paper, we call this type of approach a PICEA. Harnessing the benefits of coevolution for optimization purposes is known to be challenging [40], although there are multiobjective examples [41], and we are aware of only one existing work that has attempted to implement a concept similar to PICEA. Lohn *et al.* [42] considered coevolving a family of target vectors as a means of improving diversity across the Pareto front. The paper was published shortly before the advent of many-objective optimization in EMO and the authors did not consider the benefits of the target vectors for improving solution comparability per se. However, the paper can certainly be interpreted in such terms. The Lohn *et al.*

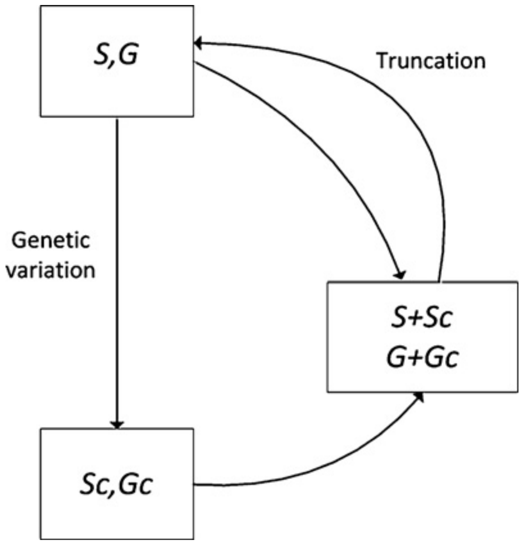


Fig. 1. (μ+λ) elitist framework.

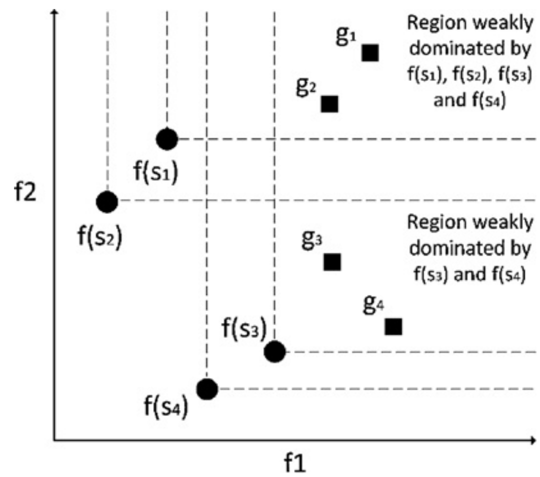


Fig. 2. Simple bi-objective minimization example.

method of fitness assignment is very interesting and we retain this in our paper on first realization of a PICEA-g.

PICEA-g considers a family of *goals*, a more natural terminology than target vectors when thinking about decision-maker preferences, but the two are essentially equivalent. It is important to recognize that this is just one realization of how preferences could be used in a PICEA, and we expect further realizations to follow once the benefits of the concept for many-objective optimization have been convincingly demonstrated.

B. Realization of a PICEA

In the fitness assignment method of Lohn *et al.* [42], candidate solutions gain fitness by meeting a particular set of target vectors in objective space, but the fitness contribution must be shared between other solutions that also satisfy those targets. Targets only gain fitness by being satisfied by a candidate solution, but the fitness is reduced the more times the targets are met by other solutions in the population. The overall aim is for the targets to adaptively guide the solution population toward the Pareto front. That is, the candidate solution population and the target population coevolve toward the Pareto front.

We develop the fitness assignment method of Lohn *et al.* into a reproducible MOEA framework as shown in Fig. 1. Fig. 1 shows a flow chart of PICEA-g within a (μ+λ) elitist framework. A population of candidate solutions and preference sets, S and G, of fixed size, N and $NGoal$, are evolved for a fixed number of generations $maxGen$. In each generation t, parents $S(t)$ are subjected to (representation-appropriate) genetic variation operators to produce N offspring $Sc(t)$. Simultaneously, $NGoal$ new preference sets $Gc(t)$, are randomly regenerated based on the initial bounds. $S(t)$ and $Sc(t)$, and $G(t)$ and $Gc(t)$, are then pooled, respectively, and the combined populations are sorted according to the fitness. Truncation selection is applied to select the best N solutions as new parent population $S(t+1)$, and $NGoal$ solutions as new preference population $G(t+1)$.

The method to calculate the fitness F_s, of a candidate solution s and fitness F_g, of a preference g, is defined by (1)–(3) as follows:

$$F_s = 0 + \sum_{g \in G \uplus G_C | s \preceq g} \frac{1}{n_g} \tag{1}$$

where n_g is the number of solutions that satisfy preference g (note that if s does not satisfy any g, then the F_S of s is defined as 0) and

$$F_g = \frac{1}{1+\alpha} \tag{2}$$

where

$$\alpha = \begin{cases} 1, & n_g = 0 \\ \frac{n_g - 1}{2N-1}, & \text{otherwise} \end{cases} \tag{3}$$

and where N is the candidate solution population size.

In order to further explain the fitness assignment scheme, consider the bi-objective minimization instance shown in Fig. 2 with two candidate solutions s_1 and s_3, their offspring s_2 and s_4, two existing preferences g_1 and g_3, and two new preferences g_2 and g_4 (i.e., $N = NGoal = 2$).

In Fig. 2, g_1 and g_2 are each satisfied by s_1, s_2, s_3, and s_4 and so $n_{g_1} = n_{g_2} = 4$. g_3 and g_4 are satisfied by s_3 and s_4 only and, therefore, $n_{g_3} = n_{g_4} = 2$. In terms of fitness of solutions, from (1)

$$F_{s_1} = F_{s_2} = \frac{1}{n_{g_1}} + \frac{1}{n_{g_2}} = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

and

$$F_{s_3} = F_{s_4} = \frac{1}{n_{g_1}} + \frac{1}{n_{g_2}} + \frac{1}{n_{g_3}} + \frac{1}{n_{g_4}} = \frac{1}{4} + \frac{1}{4} + \frac{1}{2} + \frac{1}{2} = \frac{3}{2}.$$

Considering the preference fitnesses, using (2), α for g_1 and g_2 is

$$\frac{n_{g_1} - 1}{2N - 1} = \frac{4-1}{4-1} = 1$$

and so, using (3), $F_{g_1} = F_{g_2} = \frac{1}{2}$. Similarly, α for g_3 and g_4 is $\frac{2-1}{4-1} = \frac{1}{3}$ and, therefore, $F_{g_3} = F_{g_4} = \frac{3}{4}$.

Based on the fitness, s_3 and s_4 are considered the best solutions, which will be selected for the next generation. However,

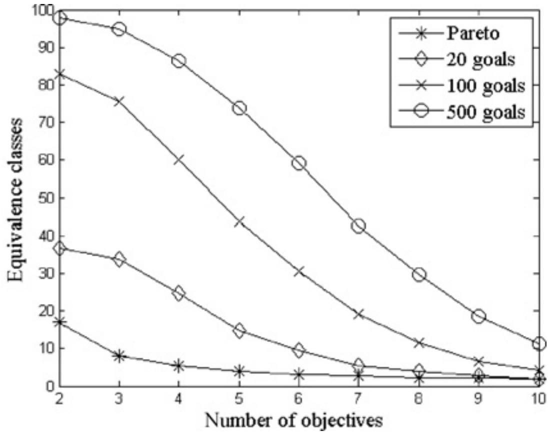


Fig. 3. Changes in comparability with objective scaling: Pareto-dominance and goal approaches.

obviously s_3 is dominated by s_4. Compared with s_3, although s_2 has a lower fitness, it is nondominated with s_4. Therefore, s_2 and s_4 are desired to be kept in the population set. In order to do that, the classic Pareto-dominance relation is incorporated. After calculating fitness values using (1)–(3), we next identify all the nondominated solutions in $S \uplus Sc$. If the number of nondominated solutions does not exceed the population size, then we assign the maximum fitness to all the nondominated solutions. However, if more than N nondominated solutions are found, we then disregard the dominated solutions prior to applying truncation selection (implicitly, their fitness is set to zero). Based on fitness, the best N nondominated solutions are selected to constitute the new parent $S(t+1)$. In the example in Fig. 2, $F_{s_1} = 0$, $F_{s_2} = \frac{1}{2}$, $F_{s_3} = 0$, and $F_{s_4} = \frac{3}{2}$.

While the use of goals does rely on Pareto-dominance comparison at the level of the individual goal, the presence of multiple goals can significantly mitigate the comparability issues observed when scaling the standard dominance relation. To see this, consider a population of 100 objective vectors randomly generated in the hypercube $(0, 1]^M$, where M is the dimension of objective space, and we have defined a direction of preference in each objective. We sort the 100 individuals into equivalence classes using a global Pareto-dominance relation (nondominated sorting) and also using the Lohn *et al.* fitness assignment scheme for three populations of randomly generated goal vectors (of size 20, 100, and 500). We repeat our experiments 500 times and calculate the mean number of equivalence classes for each of the four approaches. The results are shown in Fig. 3. It is evident that by using goals, a substantially greater level of comparability can be achieved than by using global Pareto dominance. The more goals that are used, the greater the comparability that is achieved. While the number of equivalence classes does reduce in the goal scheme as the number of objectives is increased, a 100-goal approach (i.e., matched to the number of objective vectors) is still able to provide greater comparability in ten objectives than the global Pareto approach in three objectives. This provides some reassurance that the method has potential for many-objective optimization, since Pareto-dominance-based algorithms tend to still work well for three-objective problems.

Algorithm 1 PICEA using goals (PICEA-g)

```
[S,G] = initialize(N, NGoal)
F_S = objective_function(S)
While Not termination(S)
        Sc = genetic_variation(S)
        F_Sc = objective_function(Sc)
        JointS = multiset_union(S,Sc)
        JointF = multiset_union(F_S,F_Sc)
        Gc = goal_generator(NGoal)
        JointG = multiset_union(G,Gc)
        [FitJointS, FitJointG] =
        fitness_assignment
        (JointS, JointG)
        index_nd_JointS =
        Pareto_dominance
        (JointF)
        If size (index_nd_JointS) < N
        FitJointS(index_nd_JointS) =
        max_fitness(FitJointS)
        S = truncation_selection
        (JointS, FitJointS)

    Else
        S = truncation_selection
        (JointS(index_nd_JointS),
          FitJointS(index_nd_JointS))
    End if
    G = truncation_selection
        (JointG, FitJointG)
End While
```

The pseudocode of PICEA-g is presented in Algorithm 1. The function "termination" is currently implemented as a simple maximum generation number. The goal_generator currently generates random goal vectors (variation operators have not been implemented for the goals). More specifically, goal vectors are randomly generated as objective vectors directly in objective space, within bounds defined by the vectors of ideal and anti-ideal performance.

For the comparative study, the number of preferences $NGoal$, used to evaluate candidate solutions, is set to $M \times 100$, where M is the number of objectives, i.e., 200, 400, 700, and 1000 for tests to be conducted with 2, 4, 7, and 10 objectives, respectively. Such settings may not be the best choices, and the influence of $NGoal$ on algorithm performance is discussed in Section V-A.

III. Overview of Comparison MOEAs

The performance of PICEA-g is compared with four of the best-in-class evolutionary algorithms (EAs): the ϵ-MOEA, HypE, MOEA/D, and the renowned Pareto-dominance based algorithm, NSGA-II. A fifth method, random search, is also included in the comparison as a useful baseline comparator. Each of these five algorithms is briefly summarized below, together with some details of the specific implementations adopted for the comparison.

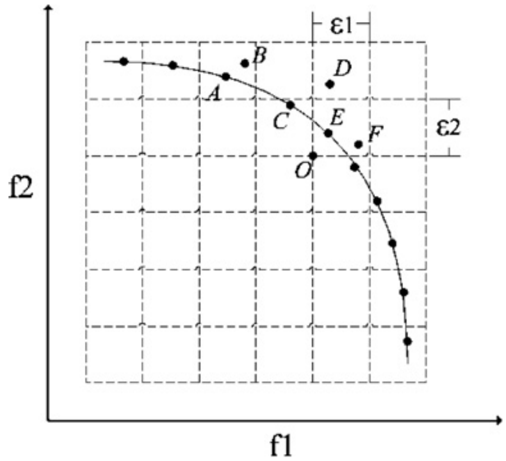


Fig. 4. Illustration of ϵ-dominance concept (minimizing $f1$ and $f2$).

A. ϵ-MOEA

Laumanns *et al.* [20] proposed the ϵ-MOEA algorithm, in which the ϵ-dominance concept is applied. Both the convergence and the diversity property of this algorithm can be maintained by the setting of an appropriate value for ϵ. The objective space is divided into a grid of hyperboxes, whose size can be adjusted by the choice of ϵ. For each hyperbox that contains a solution (or solutions), the dominance of the hyperbox is checked. An archive strategy, suggested in [34], is applied in ϵ-MOEA and used to retain one solution for each nondominated hyperbox. The specific dominance checking process is explained as follows (see Fig. 4). First, if the hyperbox of a new solution (C) dominates another hyperbox (D) in the archive, the dominated archive members (D) are rejected. Second, if there is more than one solution in the same hyperbox (A, B), the dominated solutions are removed (B). Third, if there is more than one nondominated solution in a hyperbox (E, F), one of them is randomly selected. For the third step, Deb *et al.* [34], [43] suggested choosing the solution (E) that is the closest to the origin of the hyperbox.

In this paper, Deb's ϵ-MOEA is used and, for each test instance, different ϵ[1] values are used in order to obtain roughly 100 solutions after an allowed number of function evaluations (25 000 in this paper) [34]. The value of ϵ varies with each objective, reflecting the scaling of the selected benchmark functions. We understand that the ϵ value may impact the performance of ϵ-MOEA and this is discussed in Section V-B.

B. Indicator-Based EA: HypE

Zitzler and Künzli proposed a general IBEA in [21]. Hypervolume, which has good properties with respect to set-based dominance comparisons [44], is always taken as an indicator in IBEA. However, the high computational effort required for its calculation inhibits the full exploitation of its potential [23]–[25]. HypE [27] uses a hypervolume estimation algorithm for multiobjective optimization (Monte Carlo simulation to approximate the exact hypervolume values), by which the accuracy of the estimates can be traded off against the available computing resources. There is evidence that the approach can be effective for many-objective problems [27]. In the same way as a standard MOEA, it is based on fitness assignment schemes, and consists of successive application of mating selection, variation, and environmental selection. The hypervolume indicator is applied in environmental selection.

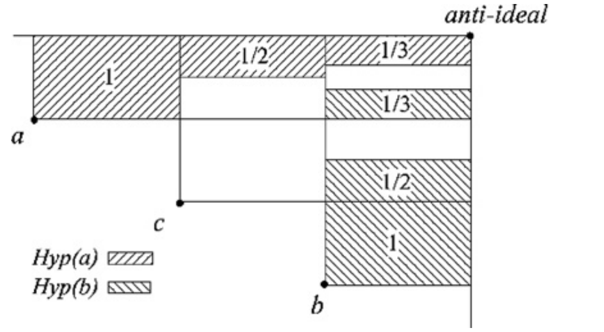


Fig. 5. Illustration of the basic fitness assignment scheme where the fitness F_a of a solution a is set to F_a = Hyp(a).

In HypE, the hypervolume-based fitness of a solution is not only calculated based on its own hypervolume contribution but also the hypervolume contribution associated with other solutions. This is illustrated in Fig. 5, where the portion of hypervolume that is weakly dominated by a is fully attributed to a, the portion of hypervolume that is dominated by a and another solution c is attributed half to a. Note that this is a more refined approach than that adopted in the other hypervolume-based approaches, such as the S metric selection-based evolutionary multi-objective optimization algorithm (SMS-EMOA) [24], in which contribution calculations are limited to single solutions, without consideration of the wider population context.

In this comparison study, we strictly follow the hypervolume contribution calculation method described in [27]. On two-objective problems, the exact hypervolume contribution of each solution is calculated. On four-, seven-, and ten-objective test instances, a Monte Carlo simulation method with 2000, 3500, and 5000 sampling points, respectively, is used to calculate the estimated hypervolume contribution. We understand that the number of sampling points *Nsp*, may impact the quality of the Pareto set approximation. A further discussion on the setting of the number of sampling points is provided in Section V-A.

C. Scalarizing Function-Based Algorithm: MOEA/D

MOEA/D, first proposed by Zhang and Li [28], is a simple yet powerful MOEA. It has a number of advantages over Pareto-dominance-based algorithms, such as its scalability to many-objective problems, high search ability for combinatorial optimization, computational efficiency of fitness evaluation, and high compatibility with local search [9], [28]–[30]. The

[1] M = 2: WFG2: ϵ = (0.004, 0.008), WFG3: ϵ = (0.0133, 0.0266), WFG4–9: ϵ = (0.02,0.04).

M = 4: WFG2: ϵ = (0.0500 0.1000 0.1500 0.2000), WFG3: ϵ = (0.1000 0.2000 0.3000 0.4000), WFG4–9: ϵ = (0.2857 0.5714 0.8571 1.1429).

M = 7: WFG2: ϵ= (0.0500 0.1000 0.1500 0.2000 0.2500 0.3000 0.3500) WFG3: ϵ= (0.2000 0.4000 0.6000 0.8000 1.0000 1.2000 1.4000) WFG4–9: ϵ = 0.4000 0.8000 1.2000 1.6000 2.0000 2.4000 2.8000).

M = 10: WFG2: ϵ=(0.0571 0.1143 0.1714 0.2286 0.2857 0.3429 0.4000 0.4571 0.5143 0.5714) WFG3: ϵ=(0.1333 0.2667 0.4000 0.5333 0.6667 0.8000 0.9333 1.0667 1.2000 1.3333) WFG4–9: ϵ= (0.5000 1.0000 1.5000 2.0000 2.5000 3.0000 3.5000 4.0000 4.5000 5.0000).

TABLE I
PARAMETERS FOR MOEA/D

Objective	Population Size (No. of Weight Vectors)	H
2	100	99
4	455	12
7	924	6
10	2002	5

main characteristic feature of MOEA/D is the handling of a multiobjective problem as a collection of single-objective problems (SOPs), which are defined by a scalarizing function (e.g., weighted sum or weighted Tchebycheff) with different weight vectors. Each scalarizing fitness function (defined by a specific weight vector) identifies a single solution that is the best with respect to that scalarizing fitness function. For each SOP, a new solution is generated by performing genetic operators on several solutions from among its neighbors. Neighbors are defined based on the distance between the weight vectors. A SOP i is a neighbor of SOP j if the weight vector of SOP i is close to that of SOP j. The newly generated solution is compared with all its neighbors. If the new solution is better, then some (or all) of its corresponding neighbors are replaced by the new solution. At the same time, the diversity of solutions is maintained by a number of uniformly distributed weight vectors in MOEA/D.

The weight vectors are generated according to

$$w_1 + w_2 + \cdots + w_M = 1 \tag{4}$$

$$w_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, \cdots \frac{H}{H}\right\}, \quad i = 1, 2, \cdots, M \tag{5}$$

where H is a user-definable positive integer and M is the number of objectives. The number of weight vectors is calculated as $N_{wv} = C_{H+M-1}^{M-1}$ [28], where C stands for the combination formula. For example, for two-objective problems, if H is specified as 100, then we can generate $C_{101}^{1} = 101$ groups of weight vectors (0, 1), (0.01, 0.99), …, (1, 0). Since each individual has a different weight vector, the population size is the same as the number of weight vectors. In the first MOEA/D version [28], the new solution will replace all the neighbors that are worse than itself. However, in order to maintain a better diversity, in [29] and [30], an upper bound is defined to limit the maximum number of replacements. In our experiments, the weighted Tchebycheff scalarizing function is used. The associated point z_i^* is specified as follows [28]:

$$z_i^* = 0.9 \times \min\{f_i(x)|x \in \Omega\}, \quad i = 1, 2, \ldots, M \tag{6}$$

where Ω is the decision space. The reference point z_i^* is updated whenever a new best value in objective i is identified. Population size and other required parameters in MOEA/D are set as shown in Table I.

Disregarding the stopping criterion (i.e., a fixed number of function evaluations), it is obvious that the more weight vectors used in MOEA/D, the better the performance that the algorithm can achieve. However, given a fixed number of function evaluations, it is not straightforward to decide how many groups of weight vectors are appropriate for each problem. In this comparison study, 100 and 455 groups of weight vectors are chosen for two and four-objective walking fish group (WFG) tests, as they are commonly used in MOEA/D studies [9], [30]. However, on seven- and ten-objective WFG tests, no related suggestions are given in the literature. The conventional method for deriving the H parameter for two- and four-objective problems is not appropriate for seven- and ten-objective cases due to the extremely large population size that results. For example, choosing $H = 12$ for seven-objective problems makes the population size as large as $C_{18}^{6} = 18564$. Therefore, we performed a limited search through the parameter space of the H parameter to find more appropriate configurations of the MOEA/D weight vectors. Our choice of H then leads to the population sizes of 924 and 2002. No specific information is provided in the literature concerning the selection of neighborhood size T and replacement neighborhood size nr except that, in [30], the authors point out that T should be much smaller than the population size and nr should be much smaller than T. For this paper, therefore, for each problem $T = 10$ and $nr = 2$. These settings may not be the best and an analysis on the setting of weight vectors T and nr is provided in Section V-C.

D. Pareto-Dominance-Based EA: NSGA-II

A wide variety of algorithms have been proposed, based on Pareto-dominance comparisons supplemented with diversity enhancement mechanisms. The most popular of these methods, the seminal NSGA-II algorithm [3], is selected in the study as representative of this class. NSGA-II is known to perform well on bi-objective problems but may experience difficulties in many-objective spaces [5], [6], [8]. It is an elitist approach; the parent and offspring population are combined and evaluated using a fast nondominated sorting approach and an efficient crowding scheme. When more than N population members of the combined population belong to the nondominated set, only those that are maximally apart from their neighbors, according to the crowding measure, are chosen.

E. Random Search: RAND

Evidence exists that random search can be competitive to evolutionary approaches in many-objective spaces [6], [14], making this a natural benchmark, at present, for comparison against any proposed new algorithm. This paper implements a very crude random scheme in which $N \times maxGen$ candidate solutions are randomly generated and the dominated solutions are removed.

IV. Test Problems, Performance Assessment, and Parameter Settings

To benchmark the performance of the considered MOEAs, problems 2–9 from the WFG test suite [45] are invoked in two-, four-, seven-, and ten-objective instances. In each case, the WFG parameters k and l are set to 18 and 14, respectively, providing a constant number of decision variables for each problem instance. Problem attributes covered include

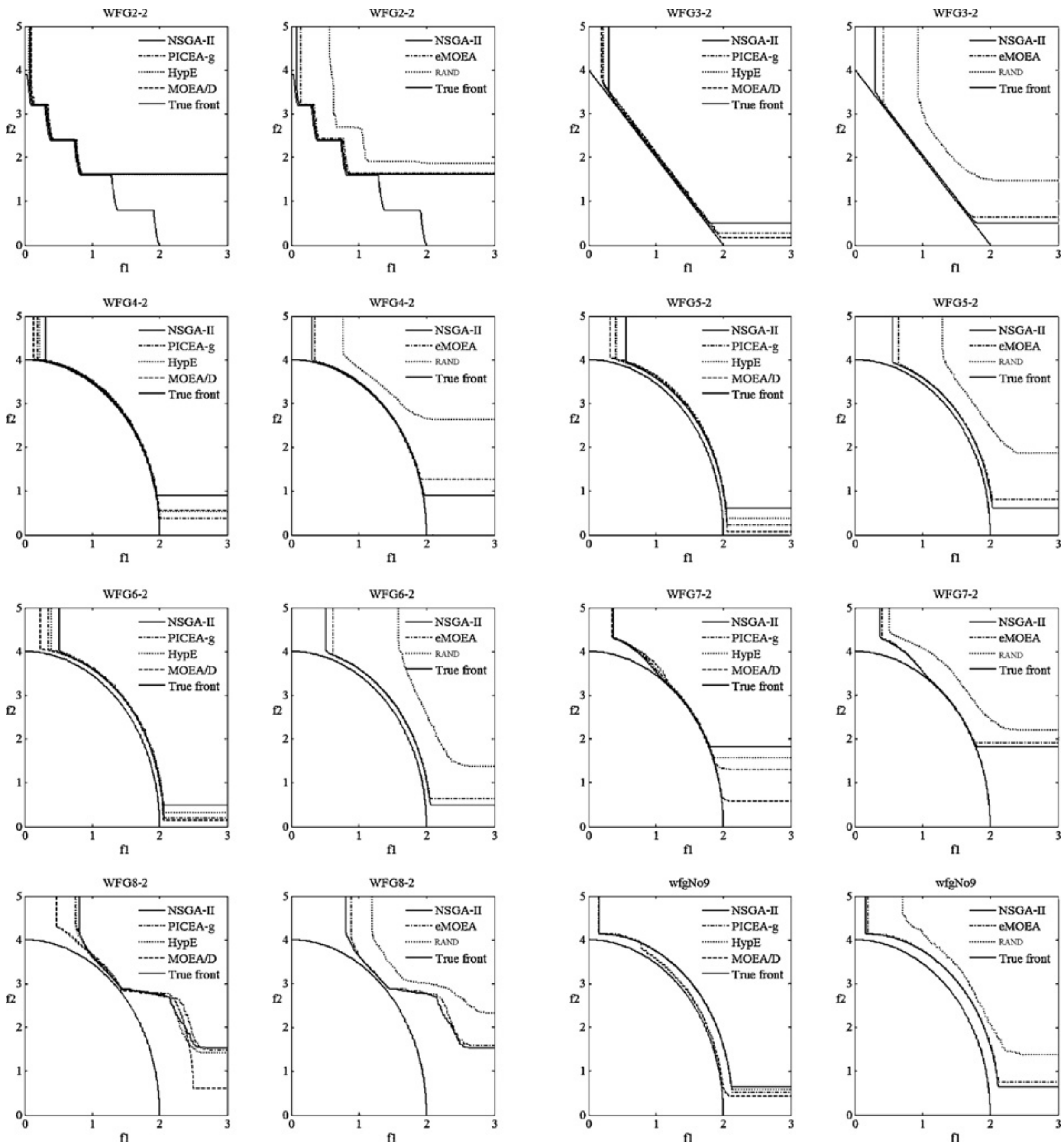


Fig. 6. Attainment surfaces for two-objective WFG test instances (WFG2-2 to WFG9-2). Color reproduction available at http://www.sheffield.ac.uk/acse/research/ecrg/picea.

separability or nonseparability, unimodality or multimodality, unbiased or biased parameters, and convex or concave geometries. Please note that in this paper, WFGX-Y refers to problem WFGX with Y objectives.

For performance assessment, first, median attainment surfaces [46] are plotted to visualize the performance of algorithms on two-objective instances. Second, comparisons are made in terms of whether one approximation set dominates another, since the only preference information this metric relies on is a direction of preference in each objective. Where such limited preference information is unable to distinguish between algorithms, approximation set comparisons are made using the hypervolume metric [22], which assumes equal relative importance of normalized objectives across the search domain. The hypervolume is calculated using the method and software developed by Fonseca *et al.* [47].

Besides some unique parameters of each algorithm (as described in Section II), the general parameters are shown in Table II. In the earlier comparative study of a PICEA [33], only mutation operators were applied in each algorithm. However, in this paper, simulated binary crossover (SBX) and polynomial mutation (PM) as described by Deb [2], [3] are applied. The recombination probability p_c is set to 1 per individual and mutation probability p_m is set to $\frac{1}{nvar}$ per decision variable. The distribution indices $\eta_c = 15$ and $\eta_m = 20$ are used. If not otherwise stated, a population size $N = 100$

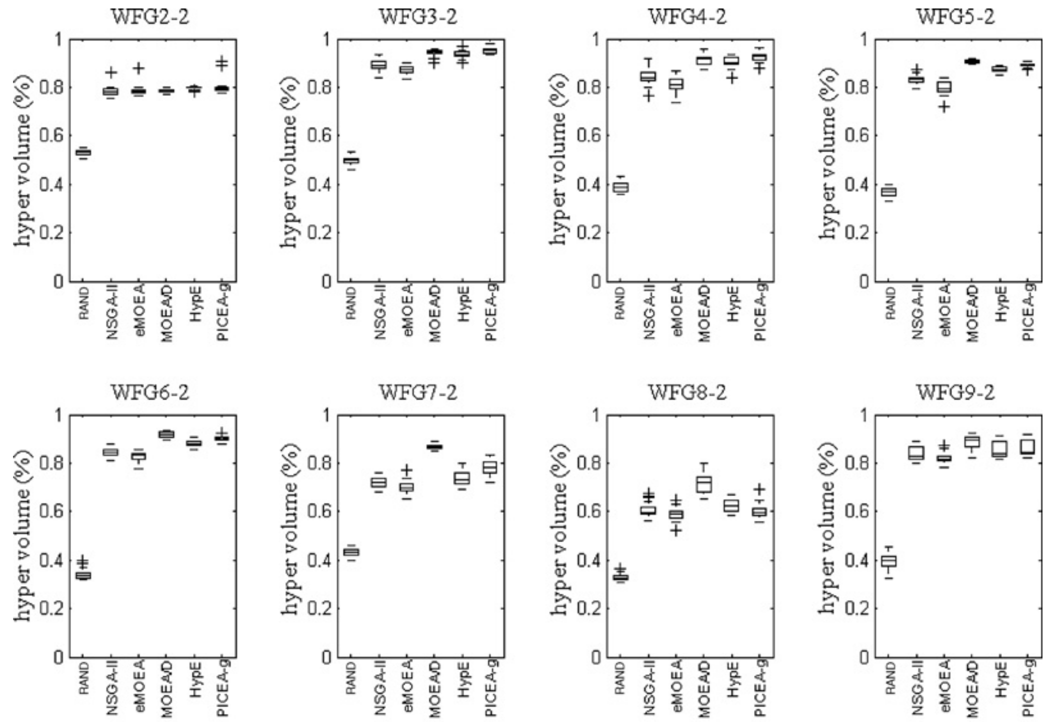


Fig. 7. Box plots of hypervolume results for two-objective instances.

TABLE II
ALGORITHM TESTING PARAMETER SETTINGS

General Parameters	
Objectives (M)	2, 4, 7, 10
Position parameter (k)	18
Distance parameter (l)	14
Decision variables ($nvar$)	$nvar = k + l = 32$
Population size (N)	100
Max generations ($maxGen$)	250
Crossover operator	SBX ($p_c = 1$, $\eta_c = 15$)
Mutation operator	PM ($p_m = \frac{1}{nvar}$, $\eta_m = 20$)

is adopted, 25 000 function evaluations are accomplished, and 31 runs of each algorithm test are performed.

V. RESULTS

In this section, we present and discuss the experimental results obtained by the algorithms, described in Section II, on the test problems identified in Section III. A comparison of the results, in terms of attainment surface, dominance rank, and hypervolume, is provided in Sections IV-A, IV-B, and IV-C, respectively.

A. Attainment Surface Results

Considering the bi-objective results as a starting point, plots of median attainment surfaces across the 31 runs of each algorithm are shown in Fig. 6. These allow visual inspection of performance in terms of the dual aims of proximity to and diversity across the global tradeoff surface [36]. For clarity, PICEA-g, HypE, NSGA-II, and MOEA/D are plotted on the left, while NSGA-II, ϵ-MOEA, and RAND are plotted on the right. (NSGA-II serves as a cross-reference.)

Qualitatively, from inspection of Fig. 6, it is clear that RAND is the worst performer. All the other MOEAs have comparable performances on convergence while different performances on diversity. Specifically, MOEA/D exhibits the best performance. PICEA-g and HypE have equivalent performances and both algorithms are slightly better than NSGA-II on all the benchmark functions. ϵ-MOEA can provide proximity as good as NSGA-II but its diversity performance is sometimes inferior. Upon closer examination, on WFG5–8, all the algorithms exhibit difficulties in converging to the true Pareto front. On WFG8, none of the algorithms considered are able to provide a good representation of the tradeoff surface. On WFG9, only MOEA/D can converge to the most part of the Pareto front. From the results of WFG5–9, we can see that bi-objective problems featuring strong multimodality or nonseparable parameters still present a challenge for best-in-class MOEAs [45].

B. Statistical Treatment

Performance comparisons between algorithms are made according to a rigorous nonparametric statistical framework, drawing on recommendations in [44]. The initial populations of candidate solutions are generated randomly for every replication of every algorithm on every problem instance and 31 replications are executed for each algorithm–instance pair. The approximation sets used in the comparisons are the members of the offline archive of all nondominated points found during the search, since this is the set most relevant to *a posteriori* decision making. For reasons of computational feasibility, prior to analysis, the set is pruned to a maximum size of 100 using

TABLE III
DOMINANCE RANKING RESULTS

Objectives	WFG	Ranking by Performance			
2	2	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	3	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	4	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	5	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	6	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	7	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	8	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	9	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
4	2	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	3	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	4	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	5	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	6	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	7	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	8	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	9	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
7	2	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND	2nd	RAND
	3	1^{st}	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	4	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	5	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	6	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	7	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	8	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	9	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
10	2	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	2nd	RAND
	3	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	4	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	5	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	6	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	7	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	8	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		
	9	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g RAND		

the SPEA2 truncation procedure [4]. The approximation sets used for performance assessment are available for download at http://www.sheffield.ac.uk/acse/research/ecrg/picea.

For each problem instance, the performance metric values (dominance rank and hypervolume) of each algorithm are calculated for each approximation set. Then the nonparametric statistical approach introduced in [48] is conducted with the performance metric values. We first test the hypothesis that all algorithms perform equally using the Kruskal–Wallis test. If this hypothesis is rejected at the 95% confidence level, we then consider pairwise comparisons between the algorithms using the Wilcoxon-ranksum two-sided comparison procedure at the 95% confidence level, employing Šidák correction to reduce Type I errors [49].

1) *Dominance Rank Results:* Dominance rank is used to evaluate the convergence of algorithms. Each approximation set for each algorithm is given an initial ranking of zero and then compared to each of the 186 approximation sets generated (31 sets for six algorithms). The rank value is incremented by one for every approximation set that weakly dominates the set being ranked. As a statistical metric, we choose the mean dominance rank of each algorithm across its 31 runs.

The Kruskal–Wallis test rejects the hypothesis that all algorithms are equivalent at the 95% confidence level. We are therefore able to consider pairwise performance comparisons between algorithms. Based on the pairwise findings, we construct partial orderings of the algorithms where, if algorithms A, B, and C are assigned to a better (i.e., lower) rank than algorithm D, this implies that all of A, B, and C have outperformed D (according to the Wilcoxon-ranksum test) on a particular problem instance (this is a similar approach to that used by Corne and Knowles [14]). Order of presentation within a partial ordering is purely alphabetical and has no performance implications. Any isolated cases of one algorithm outperforming another, but where a rank difference cannot be established (i.e., if A outperforms B, but neither A nor B can outperform C) are described separately in the text.

Using the above comparison methods, the results for dominance rank are shown in Table III. On the whole, the dominance metric, which requires the weakest assumptions about decision-maker preferences, is unable to provide any discrimination, especially on many-objective problems. In detail, at the two-objective level, random search is outperformed by all the other MOEAs. For four, seven, and ten objectives, random search performs worse than the other MOEAs only on the WFG2 problem. On all the other problems, the six considered algorithms exhibit equivalent performance in terms of dominance rank.

Considering the detailed pairwise comparisons, some differences between the algorithms are revealed. Inspecting the pairwise comparisons at the two-objective level, MOEA/D, PICEA-g, and HypE outperform NSGA-II and ϵ-MOEA, while NSGA-II performs as well as the other four MOEAs on WFG6. Similarly, on WFG9, MOEA/D exhibits better performance than ϵ-MOEA. However, the performance of PICEA-g, HypE, and NSGA-II could not be distinguished

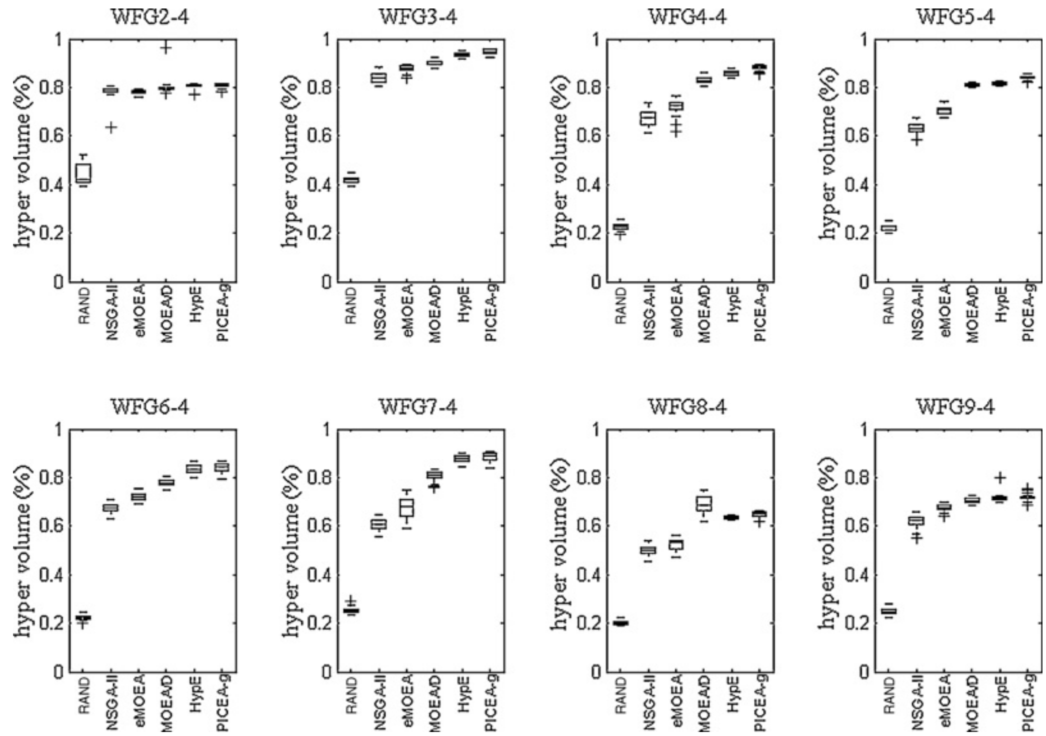


Fig. 8. Box plots of hypervolume results for four-objective instances.

from the performance of any of the other MOEAs, thereby preventing any partial ordering of the five MOEAs.

Thus, very little information can be obtained from the dominance rank comparison. A further comparison is made on the basis of the hypervolume metric.

2) *Hypervolume Results:* The reference point for calculating the hypervolume is chosen as the anti-ideal of worst possible performance in all objectives. For all WFG problems, the anti-ideal for objective i is $2i + 1$. Prior to calculating the hypervolume for an approximation set, we normalize all objectives on the range [0 1] using the anti-ideal. We also express the performance of each algorithm in terms of the proportion of the globally optimal hypervolume achieved (the method used to calculate optimal hypervolume depends on which WFG problem is being considered, and is described further in the Appendix).

The results of the Kruskal–Wallis tests followed by pairwise Wilcoxon-ranksum plus Šidák correction tests based on the hypervolume metric are provided in this section. The initial Kruskal–Wallis test breaks the hypothesis that all six algorithms are equivalent. Therefore, the outcomes of pairwise statistical comparisons for two-, four-, seven-, and ten-objective WFG problems are shown in Tables IV, V, VI, and VII, respectively. The related partial ordering of algorithms is constructed using the method previously described, again a smaller rank value indicates better performance; ordering within a rank is purely alphabetical.

Box plots [50] of Figs. 7–10 are used to visualize the distribution of the 31 hypervolume values for the associated problems. The upper and lower ends of the box are the upper and lower quartiles, while a thick line within the box encodes the median. Dashed appendages summarize the spread and shape of the distribution. Outlying values are marked as "+." The box plots allow us to consider absolute performance (in terms of the proportion of the optimal hypervolume achieved by each algorithm) in addition to the hypothesis testing around relative performance.

From the statistical comparison results in Table IV and box plots in Fig. 7, the following key points can be observed for two-objective WFG problems.

1) MOEA/D is always among the top performing algorithms except for WFG2, where HypE and PICEA-g are the best performers.
2) PICEA-g outperforms ϵ-MOEA and NSGA-II on all the benchmarks except for WFG8 and WFG9, where the three algorithms perform comparably.
3) HypE is inferior to PICEA-g on WFG3, 5–7, while it performs better on WFG8 and comparably on WFG9.
4) The performance of ϵ-MOEA is worse than (problems WFG3–7) or at best equivalent to (WFG2, 8, and 9) other MOEAs.
5) NSGA-II exhibits mixed performance. It is ranked in the second class on WFG2 and WFG9, but it is worse than PICEA-g, HypE, and MOEA/D on problems WFG3–6. Its performance is equivalent to PICEA-g, ϵ-MOEA on WFG8 and WFG9. Also, it performs worse than (WFG8) or comparably to (WFG7 and WFG9) HypE.
6) All five MOEAs outperform random search.

It is clear from Table V and Fig. 8 that for all four-objective WFG problems, all MOEAs outperform random search. The performance of PICEA-g remains promising. In detail, except for WFG8 (MOEA/D is the best), PICEA-g is ranked first on all benchmark functions, exclusively for WFG3–5, and jointly with HypE on WFG2, 6, 7, and 9. Reinforcing conclusions

TABLE IV

HYPERVOLUME RESULTS FOR TWO-OBJECTIVE INSTANCES

WFG		Ranking by Performance	WFG		Ranking by Performance
2	1st	HypE PICEA-g	3	1st	MOEA/D PICEA-g
	2nd	ϵ-MOEA MOEA/D NSGA-II		2nd	HypE
	3rd	RAND		3rd	NSGA-II
				4th	ϵ-MOEA
				5th	RAND
4	1st	HypE MOEA/D PICEA-g	5	1st	MOEA/D
	2nd	NSGA-II		2nd	PICEA-g
	3rd	ϵ-MOEA		3rd	HypE
	4th	RAND		4th	NSGA-II
				5th	ϵ-MOEA
				6th	RAND
6	1st	MOEA/D	7	1st	MOEA/D
	2nd	PICEA-g		2nd	PICEA-g
	3rd	HypE		3rd	HypE NSGA-II
	4th	NSGA-II		4th	ϵ-MOEA
	5th	ϵ-MOEA		5th	RAND
	6th	RAND			
8	1st	MOEA/D	9	1st	MOEA/D
	2nd	HypE		2nd	ϵ-MOEA HypE NSGA-II PICEA-g
	3rd	ϵ-MOEA NSGA-II PICEA-g		3rd	RAND
	4th	RAND			

TABLE V

HYPERVOLUME RESULTS FOR FOUR-OBJECTIVE INSTANCES

WFG		Ranking by Performance	WFG		Ranking by Performance
2	1st	HypE PICEA-g	3	1st	PICEA-g
	2nd	MOEA/D NSGA-II		2nd	HypE
	3rd	ϵ-MOEA		3rd	MOEA/D
	4th	RAND		4th	NSGA-II
				5th	ϵ-MOEA
				6th	RAND
4	1st	PICEA-g	5	1st	PICEA-g
	2nd	HypE		2nd	HypE
	3rd	MOEA/D		3rd	MOEA/D
	4th	ϵ-MOEA		4th	ϵ-MOEA
	5th	NSGA-II		5th	NSGA-II
	6th	RAND		6th	RAND
6	1st	HypE PICEA-g	7	1st	HypE PICEA-g
	2nd	MOEA/D		2nd	MOEA/D
	3rd	ϵ-MOEA		3rd	ϵ-MOEA
	4th	NSGA-II		4th	NSGA-II
	5th	RAND		5th	RAND
8	1st	MOEA/D	9	1st	HypE PICEA-g
	2nd	PICEA-g		2nd	MOEA/D
	3rd	HypE		3rd	ϵ-MOEA
	4th	ϵ-MOEA		4th	NSGA-II
	5th	NSGA-II		5th	RAND
	6th	RAND			

from previous studies, the Pareto-dominance-based NSGA-II begins to struggle on four-objective problems; it performs equivalently to MOEA/D on WFG2 and better than ϵ-MOEA on WFG3, but exhibits worse performance than both MOEA/D and ϵ-MOEA on the remaining benchmark functions.

As the number of objectives increases to 7, we can observe (see Table VI and Fig. 9) that PICEA-g and HypE outperform the other three MOEAs. Interestingly, random search still performs least well in all cases. Upon closer examination, the findings are as follows.

1) PICEA-g and HypE are ranked first and second for WFG4–8, respectively. The two algorithms are jointly ranked in the first class on WFG2, 3, and 9.
2) MOEA/D performs better than (on WFG3, 5, and 7) or at least equivalent to (on WFG2, 4, 6, and 9) ϵ-MOEA on all the benchmark functions. NSGA-II exhibits an inferior performance to ϵ-MOEA on all tests, except WFG2 and WFG8, where it gives comparable performance with MOEA/D and ϵ-MOEA.
3) All the MOEAs have a comparable performance for WFG2, where absolute coverage of the globally optimal hypervolume remains at over 80%.

Results for 10-objective WFG problems are shown in Table VII and Fig. 10. Considering relative comparisons, random search continues to perform badly and the superiority of PICEA-g and HypE is more established. The performances of these two algorithms are statistically better than all the other algorithms on all benchmark functions except for WFG2, on which the five MOEAs are all in the first class (as for WFG2 with seven objectives) and are better than random

TABLE VI
HYPERVOLUME RESULTS FOR SEVEN-OBJECTIVE INSTANCES

WFG		Ranking by Performance	WFG		Ranking by Performance
2	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	3	1st	HypE PICEA-g
	2nd	RAND		2nd	MOEA/D
				3rd	ϵ-MOEA
				4th	NSGA-II
				5th	RAND
4	1st	PICEA-g	5	1st	PICEA-g
	2nd	HypE		2nd	HypE
	3rd	ϵ-MOEA MOEA/D		3rd	MOEA/D
	4th	NSGA-II		4th	ϵ-MOEA
	5th	RAND		5th	NSGA-II
				6th	RAND
6	1st	PICEA-g	7	1st	PICEA-g
	2nd	HypE		2nd	HypE
	3rd	ϵ-MOEA MOEA/D		3rd	MOEA/D
	4th	NSGA-II		4th	ϵ-MOEA
	5th	RAND		5th	NSGA-II
				6th	RAND
8	1st	PICEA-g	9	1st	HypE PICEA-g
	2nd	Hypϵ-MOEAE		2nd	ϵ-MOEA MOEA/D
	3rd	ϵ-MOEA MOEA/D NSGA-II		3rd	NSGA-II
	4th	RAND		4th	RAND

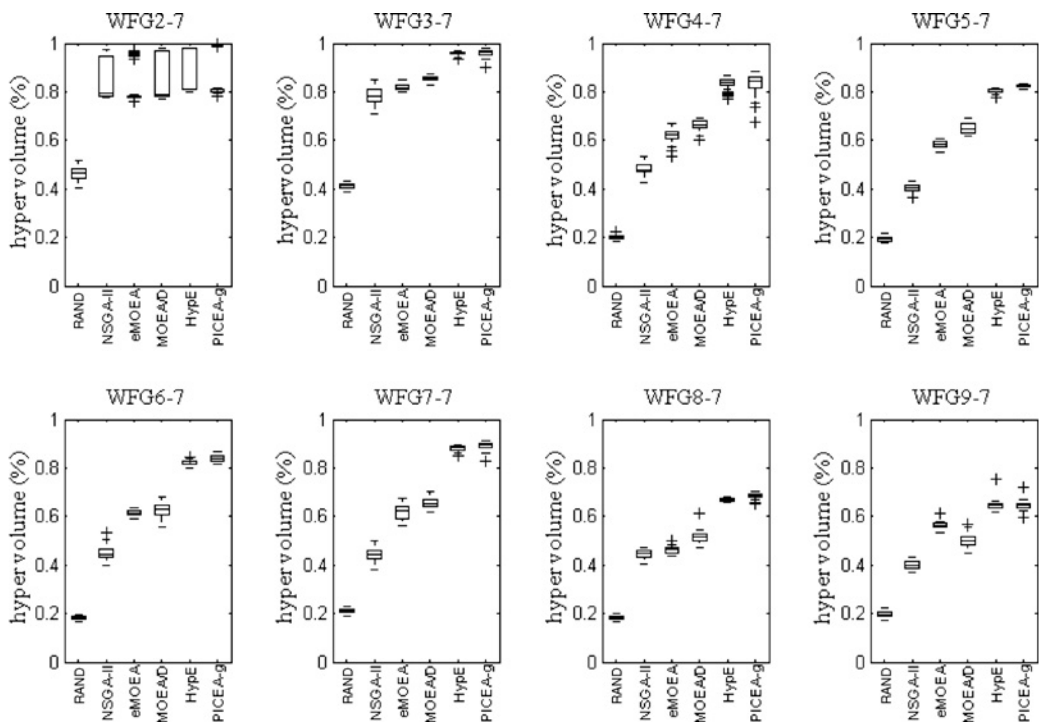


Fig. 9. Box plots of hypervolume results for seven-objective instances.

search. Specifically, PICEA-g and HypE jointly rank in the first class for WFG3, 4, 7, and 8. PICEA-g is the exclusive best for WFG5 and WFG6, and, likewise, HypE the exclusive best for WFG9. Among MOEA/D, ϵ-MOEA, and NSGA-II, MOEA/D exclusively gives the best performance on WFG4, 5, 7, and 8. ϵ-MOEA is somewhat inferior to MOEA/D on most of the benchmark functions, however, for WFG6 and 9, it outperforms MOEA/D. The performance of both algorithms is comparable on WFG3. NSGA-II performs worst on all benchmark functions except for WFG8, where ϵ-MOEA is the worst. In terms of absolute performance, the box plots show that on all problems except WFG8 and WFG9, HypE and PICEA-g are still able to achieve over 80% of the global hypervolume value. A crude random search can achieve approximately 20% coverage of the optimal hypervolume with all MOEAs performing better than this for the equivalent number of candidate solution evaluations.

C. Supplementary Results

To further understand the performance of the algorithms, we have also separately calculated proximity (as measured by generational distance (GD) [51]) and diversity (as measured by the spread metric [2]) measures for the seven-objective WFG4–9 benchmark functions. The true Pareto front for these

TABLE VII

HYPERVOLUME RESULTS FOR 10-OBJECTIVE INSTANCES

WFG		Ranking by Performance	WFG		Ranking by Performance
2	1st	ϵ-MOEA HypE MOEA/D NSGA-II PICEA-g	3	1st	HypE PICEA-g
	2nd	RAND		2nd	ϵ-MOEA MOEA/D
				3rd	NSGA-II
				4th	RAND
4	1st	PICEA-g HypE	5	1st	PICEA-g
	2nd	MOEA/D		2nd	HypE
	3rd	ϵ-MOEA NSGA-II		3rd	MOEA/D
	4th	RAND		4th	ϵ-MOEA
				5th	NSGA-II
				6th	RAND
6	1st	PICEA-g	7	1st	HypE PICEA-g
	2nd	HypE		2nd	MOEA/D
	3rd	ϵ-MOEA		3rd	ϵ-MOEA
	4th	MOEA/D		4th	NSGA-II
	5th	NSGA-II		5th	RAND
	6th	RAND			
8	1st	HypE PICEA-g	9	1st	HypE
	2nd	MOEA/D		2nd	PICEA-g
	3rd	NSGA-II		3rd	ϵ-MOEA
	4th	ϵ-MOEA		4th	MOEA/D
	5th	RAND		5th	NSGA-II
				6th	RAND

problems is a regular geometric shape that is amenable to uniform sampling. We sample 20 000 points as the reference set for calculating the metrics. The statistical tests are based on the mean values of the performance indicators, and the same nonparametric procedures are adopted as earlier. The GD and spread metric results are shown in Tables VIII and IX, respectively.

MOEA/D is found to achieve the best proximity on four of the six problems considered; however, it tends to rank quite poorly in terms of diversity. NSGA-II is found to provide inferior proximity to random search on all six benchmark problems, but with a diversity metric that is superior to random. PICEA-g consistently ranks among the top two for both proximity and spread, and is the only algorithm under test to achieve such a performance.

VI. DISCUSSION

The empirical comparison has identified that, from the representative algorithms considered, PICEA-g and HypE are presently good options for many-objective optimization. Meanwhile, MOEA/D exhibits outstanding performance on bi-objective problems, but has not performed so well in a many-objective context.

While NSGA-II was seen to outperform random search in many-objective spaces according to hypervolume, further interrogation of the seven-objective results has confirmed that this tends to be based on approximation sets with equivalent or worse proximity to random search, yet retaining good diversity. This result was first identified in [5] and confirmed in [8] and is believed to be due to dominance resistance (in this case, due to many objectives) coupled with an active diversity promotion mechanism that favors remote solutions far away from the global Pareto front. Having said this, NSGA-II was able to perform quite well in absolute terms on the many-objective WFG2 benchmark functions. It is known that standard Pareto-dominance-based approaches can perform well when the dimensionality of the Pareto front is not many objective [52] or if the objectives are highly correlated [53], [54], but these are not characteristics of the WFG2 problem. It remains unclear why NSGA-II can perform well in this many-objective space, but understanding this issue may well unlock further understanding of MOEA performance.

Despite the unremarkable performance of MOEA/D on many-objective problems, according to hypervolume, the seven-objective results focusing separately on proximity and spread show that this algorithm is still very capable at finding solutions that are close to the global Pareto front. The issue is a loss of diversity, which is likely to be due to inappropriate specifications of *a priori* weight vectors, itself arising from a general lack of knowledge in the literature about how to configure the algorithm in many-objective spaces. We explore this configuration issue further below, as part of a wider discussion embracing all the algorithms.

As explained in Section II, PICEA-g, HypE, MOEA/D, and ϵ-MOEA all have some unique parameters beyond those normally found in EAs—e.g., $NGoal$ in PICEA-g, Nsp in HypE, ϵ in ϵ-MOEA, and the weight vectors T and nr in MOEA/D. In the previous sections, we used standard parameter settings from the literature. In this section, we will provide some analysis of the influence of these parameters for the algorithms, to provide a level of validation to our findings.

A. Influence of Parameter Settings for PICEA-g and HypE

The comparison study has identified that, in terms of hypervolume, PICEA-g and HypE tend to perform much better than the other EAs on all of the many-objective WFG tests. The related parameters $NGoal$ is set to 200, 400, 700, and 1000 for two-, four-, seven-, and ten-objective problems, respectively, and Nsp is set to 2000, 3500, and 5000 for four-, seven-,

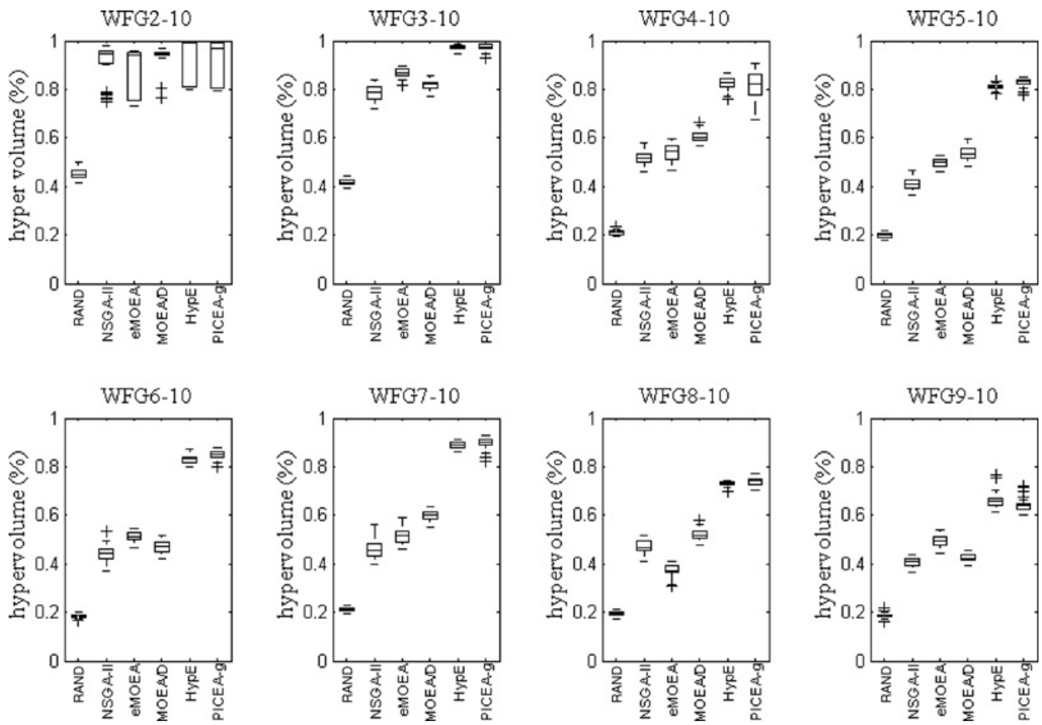


Fig. 10. Box plots of hypervolume results for 10-objective instances.

TABLE VIII
GD RESULTS FOR SELECTED SEVEN-OBJECTIVE INSTANCES

WFG		Ranking by GD	WFG		Ranking by GD
4	1st	MOEA/D	5	1st	MOEA/D
	2nd	PICEA-g		2nd	PICEA-g
	3rd	HypE		3rd	HypeE
	4th	ϵ-MOEA		4th	ϵ-MOEA
	5th	RAND		5th	RAND
	6th	NSGA-II		6th	NSGA-II
6	1st	MOEA/D	7	1st	PICEA-g
	2nd	PICEA-g		2nd	HypE MOEA/D
	3rd	HypE		3rd	ϵ-MOEA
	4th	ϵ-MOEA		4th	RAND
	5th	RAND		5th	NSGA-II
	6th	NSGA-II			
8	1st	PICEA-g	9	1st	MOEA/D
	2nd	HypE		2nd	HypE PICEA-g
	3rd	MOEA/D RAND		3rd	ϵ-MOEA
	4th	ϵ-MOEA		4th	RAND
	5th	NSGA-II		5th	NSGA-II

and ten-objective problems, respectively (on two-objective problems, the exact hypervolume value is used). However, such settings perhaps are not the best options. A simple analysis is provided in this section. Tables X and XI provide alternative settings for $NGoal$ and Nsp for different problems and Figs. 11 and 12 present representative experimental results arising from these different settings when the algorithms are applied to four-objective problems.

Fig. 11 clearly illustrates that the larger the number of goal vectors used, the better the performance that PICEA-g can deliver. However, the improvement is not linear. Similarly, from Fig. 12, it is observed that the greater the samples are used, the better the quality of Pareto set approximation that can be obtained by HypE; these HypE results are similar to those obtained in [27]. Additionally, for both algorithms, it is obvious that as the size of $NGoal$ or Nsp increases, there is a corresponding increase in the resulting computational cost. Clearly, choice of a suitable value for $NGoal$ or Nsp demands an appreciation of the tradeoff between computational effort and performance.

Although only results on four-objective WFG problems are provided above, similar experiments were performed on WFG problems with seven and ten objectives, with similar outcomes.

B. Influence of Parameter Settings for ϵ-MOEA

On most many-objective problems, ϵ-MOEA can outperform NSGA-II. However, for WFG8-10, it performs worse

TABLE IX

SPREAD METRIC RESULTS FOR SELECTED SEVEN-OBJECTIVE INSTANCES

WFG		Ranking by Spread	WFG		Ranking by Spread
4	1st	HypE	5	1st	HypE
	2nd	ϵ-MOEA PICEA-g		2nd	ϵ-MOEA PICEA-g
	3rd	NSGA-II		3rd	NSGA-II
	4th	MOEA/D		4th	MOEA/D
	5th	RAND		5th	RAND
6	1st	ϵ-MOEA HypE PICEA-g	7	1st	HypE
	2nd	NSGA-II		2nd	PICEA-g
	3rd	MOEA/D		3rd	ϵ-MOEA
	4th	RAND		4th	NSGA-II
				5th	MOEA/D
				6th	RAND
8	1st	ϵ-MOEA MOEA/D NSGA-II	9	1st	ϵ-MOEA HypE
	2nd	HypE PICEA-g		2nd	PICEA-g
	3rd	RAND		3rd	NSGA-II
				4th	MOEA/D
				5th	RAND

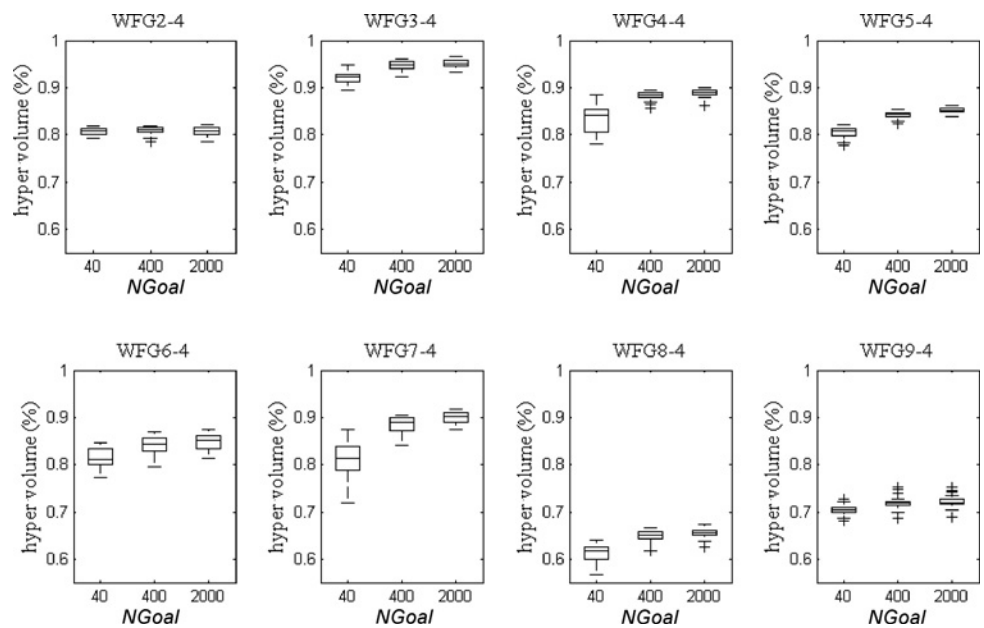


Fig. 11. Performance of PICEA-g on four-objective problems, for different numbers of goal vectors.

TABLE X

PARAMETER SETTINGS OF $NGoal$

	No. of Goal Vectors ($NGoal$)		
Objectives (M)	$M \times 10$	$M \times 100$	$M \times 500$
2	20	200	1000
4	40	400	2000
7	70	700	3500
10	100	1000	5000

TABLE XI

PARAMETER SETTINGS OF Nsp

Objectives (M)	No. of Sampling Points (Nsp)		
	$M \times 250$	$M \times 500$	$M \times 1000$
4	1000	2000	4000
7	1750	3500	7000
10	2500	5000	10 000

than NSGA-II. The reason for this may be an inappropriate parameter setting for ϵ. In this paper, ϵ values were selected for different test problems in such a way that there are around 100 solutions in the final approximate set S. However, it is always difficult to specify the correct ϵ value for each problem. In this section, the influence of ϵ values on algorithm performance is studied. WFG4 with four objectives is taken as an instance. Table XII lists three groups of ϵ values. Fig. 13 illustrates how the hypervolume value (averaged over 31 algorithm runs) for the three settings changes during the evolutionary process.

TABLE XII

ϵ-SETTINGS FOR WFG4-4 PROBLEMS

	$\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$
1	$\epsilon = 0.2, 0.4, 0.6, 0.8$
2	$\epsilon = 0.001, 0.002, 0.003, 0.004$
3	$\epsilon = 1, 2, 3, 4$

From Fig. 13, when ϵ-MOEA is run with large ϵ values it does not perform well because few solutions can be stored in the archive. Alternatively, when ϵ-MOEA is run with small

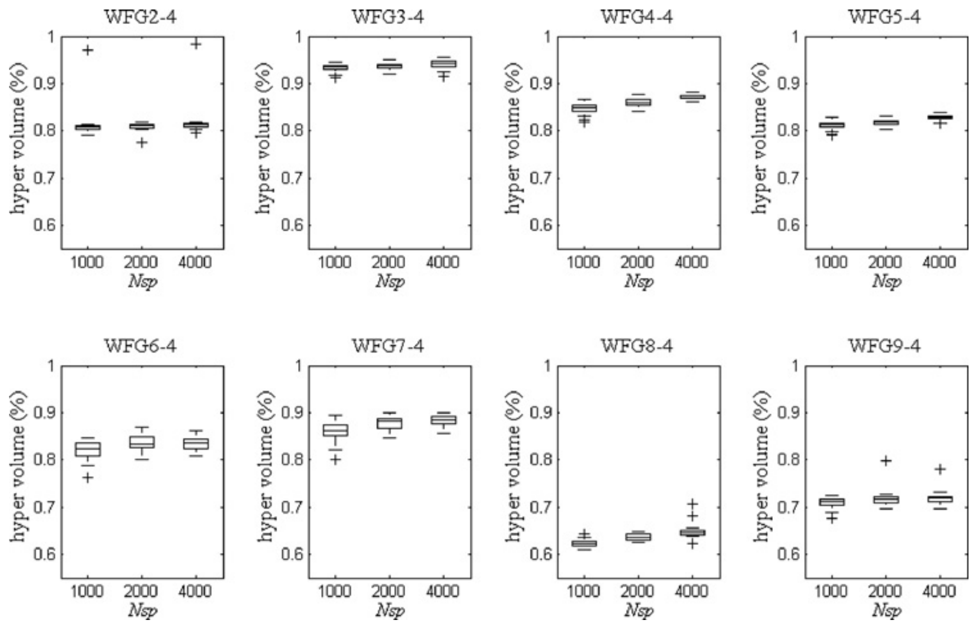


Fig. 12. Performance of HypE on four-objective problems, for different numbers of sampling points.

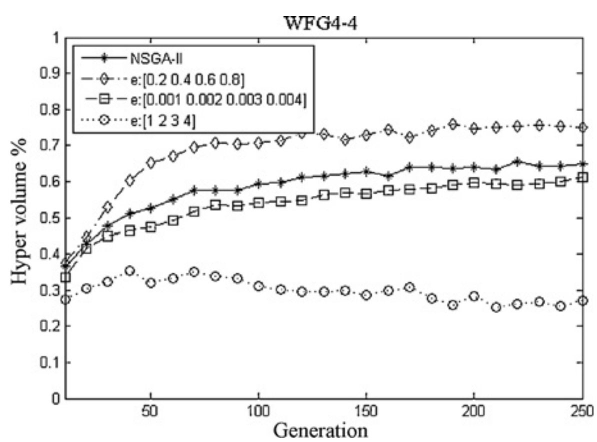


Fig. 13. Performance of ϵ-MOEA and NSGA-II on four-objective WFG4.

ϵ values, its performance degrades to be similar to that of NSGA-II, for the reason that no significant difference exists between the ϵ-dominance relation and the Pareto-dominance relation. A similar performance was observed for other WFG problems.

C. *Influence of Parameter Settings for MOEA/D*

In the comparison study, MOEA/D did not perform as well as we expected. The reason for this might be due to the parameter configurations of evenly distributed weight vectors and selection or replacement neighbor size.

In a similar way to the original set of experiments described in Section III, 25 000 function evaluations are taken as the stopping criterion, the same variation operators are applied, but this time different weight vectors are used. Each case is executed 31 times. WFG problems with seven and ten objectives are chosen as the study cases. Table XIII presents the number of weight vectors used in this test. Fig. 14 presents box plots of the hypervolume results; random search serves as a baseline reference method.

TABLE XIII
WEIGHT VECTORS SETTINGS ON SEVEN AND TEN-OBJECTIVE WFG INSTANCES

Objectives (M)	No. of Weight Vectors/H			
7	100[a]	462/5	924/6	1716/7
10	100[b]	715/4	2002/5	5005/6

[a]Uniformly select 100 weight vectors from weight vector set 924 (H = 6).
[b]Uniformly select 100 weight vectors from weight vector set 2002 (H = 5).

Fig. 14 shows that setting the number of weight vectors as 924 and 2002 for the 7- and 10-objective tests, respectively, is comparatively a good choice. Moreover, MOEA/D with 100 weight vectors gives the worst performance on all the test problems; it cannot even significantly outperform random search on some ten-objective tests. From the experimental results, it is evident that the performance of MOEA/D varies significantly, depending on the choice of the number of weight vectors. The selection of the appropriate number of weight vectors to be used demands further study.

Turning to the influence of selection neighbor size T and replacement neighborhood size nr, 20 groups of T and nr settings are examined on WFG4 with 7 and 10 objectives, respectively (see Table XIV). The average hypervolume values arising from 31 runs for each setting are shown in Fig. 15. The symbol "$\lceil\rceil$" returns the value of a number rounded up to the nearest integer.

It is obvious that the performance of MOEA/D can be affected considerably by the choice of T and nr. A similar conclusion is also found in [9] and [55]. As for the setting used in the comparison study, for seven-objective problem, $T = \lceil 924 \times 1\% \rceil = 10, nr = \lceil 10 \times 20\% \rceil = 2$ is the best setting, and for 10-objective problem, $T = \lceil 2002 \times 0.5\% \rceil = 10, nr = \lceil 10 \times 20\% \rceil = 2$ is almost the best setting as well. In

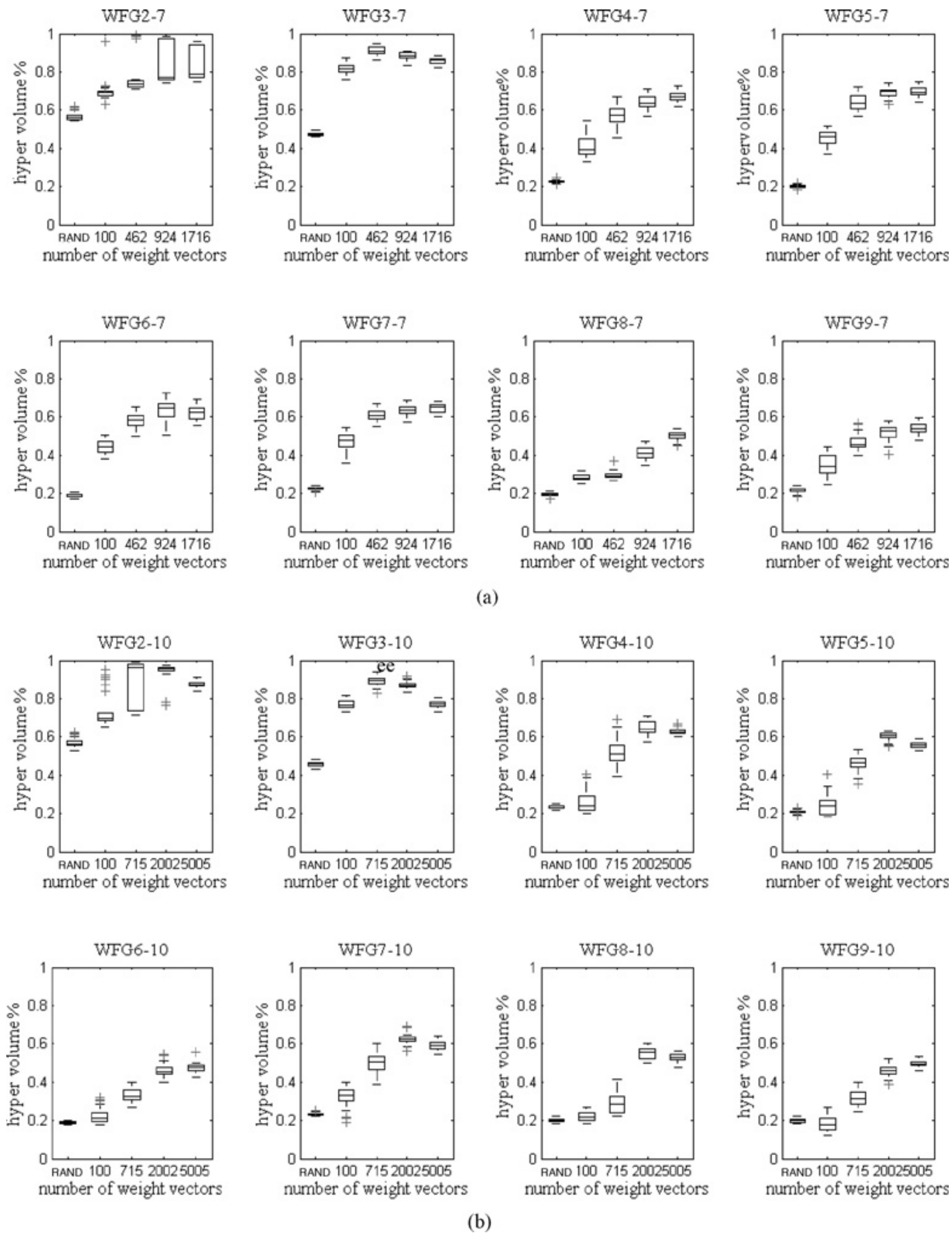


Fig. 14. Box plots of hypervolume results. (a) WFG2–9 for seven objectives. (b) WFG2–9 for 10 objectives.

summary, we observe that MOEA/D is sensitive to parameter settings, such as the number of weight vectors, selection, and replacement neighbor size. Therefore, it is desirable to choose appropriate parameter values or perhaps introduce adaptive parameter control strategies to adjust parameter values while using MOEA/D.

Although we focused on the parameters of the MOEA/D, it is recognized that the choice of scalarizing function itself is an important consideration. Ishibuchi *et al.* identified that the standard implementation of weighted Tchebycheff scalarizing function was often outperformed by the weighted sum scalarizing function, the augmented weighted Tchebycheff scalarizing function, and combined use of both these functions [56].

VII. Findings, Limitations, and Future Research

A. Findings

We have carried out a systematic comparison of six algorithms that included five different classes of MOEAs: a PICEA-g, a Pareto-dominance-based algorithm (NSGA-II), an ϵ-dominance-based algorithm (ϵ-MOEA), a scalarizing function-based algorithm (MOEA/D), and an indicator-based algorithm (HypE). The main findings are as follows.

1) A particular concern with optimization-focused implementations of coevolution is the potential for pathologies, such as the Red Queen effect (subjective fitness improves without any corresponding improvement in objective fitness or vice versa), cycling

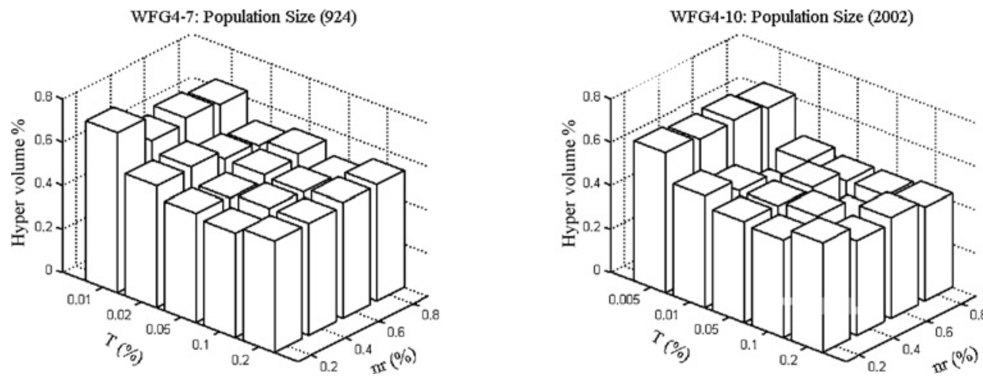


Fig. 15. Hypervolume results on 7- and 10-objective WFG4.

TABLE XIV
MOEA/D CONFIGURATION (T AND nr) FOR WFG4

No. of objectives (M)	7	10
Parameter (H) and subsequent population size (N)	$H = 6 \Rightarrow N = 924$	$H = 5 \Rightarrow N = 2002$
Selection neighborhood size (T) as % of N	$T \in$ (0.01 0.02 0.05 0.1 0.2)	$T \in$ (0.005 0.01 0.05 0.1 0.2)
Competition neighbor size (nr) as % of T	$nr \in$ (0.2 0.4 0.6 0.8)	$nr \in$ (0.2 0.4 0.6 0.8)

(subjective fitness exhibits limit cycle dynamics, without incremental improvement), and disengagement (loss of fitness discrimination due to the total superiority of one population) [40], [57]. Fortunately, the fitness assignment scheme in PICEA-g, based on the approach of Lohn *et al.*, appears resistant to these issues; as anticipated, both the candidate solutions and the preferences can converge toward the Pareto-optimal front. Fig. 16 shows how hypervolume (normalized by the true hypervolume value) changes over the course of the evolution for candidate solutions and preferences. Note that the preference dynamics tend to slightly lag the solutions, so the "greyhound" analogy proposed by Klee and Lamont [41] is not strictly accurate. The reason for the lag is that preferences that cannot be met by any solutions are assigned the worst possible fitness score in the Lohn *et al.* scheme.

2) The PICEA-g exhibits promising performance for many-objective problems. It is found to be consistently among the best algorithms across the test problems considered. In addition to superior performance, as measured by the hypervolume indicator, on many-objective problems, it also offers competitive performance with the popular NSGA-II in bi-objective environments.
3) The estimated hypervolume IBEA (HypE) gives very competitive performance on both bi-objective and many-objective problems. Its performance is close to PICEA-g and better than other MOEAs on most of the selected benchmarks.
4) The larger the number of goal vectors $NGoal$ used in PICEA-g, the better the algorithm performs. Similarly, the larger the number of sampling points Nsp used in HypE, the better the algorithm performs. Of course, for both of these increases, there is a corresponding increase in computational cost.
5) The concept of ϵ-dominance is much more effective than pure Pareto dominance in solving many-objective problems. Although ϵ-MOEA does not exhibit the best performance, it outperforms NSGA-II on most of the many-objective problems studied. However, the hypervolume measure of ϵ-MOEA is not found to be better than NSGA-II on bi-objective WFG tests. This may be due to the fact that ϵ-MOEA is not effective in obtaining extreme solutions on the Pareto front [34], [43] or perhaps the ϵ value is set inappropriately.
6) MOEA/D performs well on two-objective problems. However, its performance is not particularly notable on the many-objective problems studied. This might be explained, partly, by the parameter settings used. The results of Section V-C demonstrate that MOEA/D is sensitive to parameters such as the number of weight vectors and the selection or replacement neighbor size.
7) The classical Pareto dominance and density-based algorithm (NSGA-II) can perform well on bi-objective problems. However, its performance is significantly degraded, in both relative and absolute terms, when dealing with many-objective problems.
8) In terms of hypervolume, all the MOEAs considered offer better performance than a crude random strategy. Note that other studies have found that MOEAs can degenerate to random search (or possibly worse) on many-objective problems.

B. Limitations and Future Research

The study has two main limitations. The first is that although the same number of function evaluations are executed for all of the algorithms, the population size used in MOEA/D is necessarily different from the other algorithms, which may induce some bias. The second limitation is the HypE algorithm uses hypervolume to guide the optimization, yet we have used this metric as a performance indicator (since it has good properties) and this might be deemed unfair in relation to other algorithms. A further minor point to consider is the suitability of the choice of 25 000 function evaluations as a stopping criterion. It might

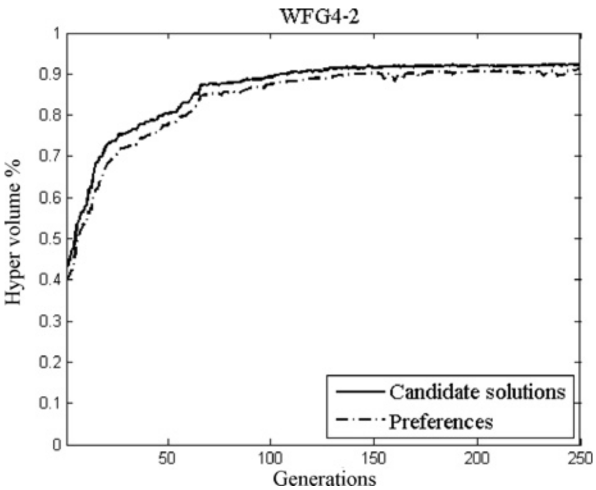


Fig. 16. Illustration of the performance of candidate solutions and preferences on bi-objective WFG4.

be better to consider alternative termination criteria such as hypervolume gradient values, although plots of convergence according to hypervolume (not shown) suggested that reasonable convergence was achieved within the limit by all algorithms (incremental improvement rates appeared, qualitatively, small).

With respect to further research, it is our view that first, there needs to be a systematic analysis of how performance varies with the tuneable parameters of the algorithms [58], [59]. This kind of analysis is important to gain insight into: 1) the robustness of the algorithms and to mitigate the possibility of misleading findings; and 2) the absolute potential of each method when tuned to the problem. Second, adaptive parameter control strategies should be investigated; such strategies have proved beneficial for the performance of EAs [60]–[62]. Third, the goal vector approach examined in this paper represents just one possible formulation of the preference-inspired coevolutionary concept and further research into other realizations is warranted [42], [33]. Within the existing goal vector formulation, the requirement to determine the bounds on the space of goals is a limitation of the method. We have used known, problem-specific, ideal and anti-ideal vectors as bounds; in practice, these would need to be estimated, either via expert domain-specific knowledge or via preliminary single-objective optimizations. Random generation within such bounds is arguably inefficient because some goal vectors may be infeasible and also evolving problem knowledge is not exploited. Potentially, the goal generator could include genetic variation operators to improve search efficiency. Fourth, our findings are based on real-parameter function optimization problems, but it is also important to assess EMO algorithms on other problem types, such as many-objective combinatorial problems, and also, crucially, real-world problems.

VIII. Conclusion

This paper proposed a new concept for solving many-objective optimization problems: PICEAs. In the algorithms, a family of preferences were coevolved with candidate solutions, the preferences gain higher fitness by being satisfied by fewer candidate solutions, and the candidate solutions gain fitness by meeting as many preferences as possible. We realized a specific algorithm based on this concept, PICEA-g, and compared it rigorously to four best-in-class MOEAs (NSGA-II, ϵ-MOEA, MOEA/D, and HypE) and a random search benchmark on the leading WFG test problems with two, four, seven, and ten objectives. According to the empirical results, PICEA-g showed highly competitive performance and can therefore make a strong claim for use on many-objective problems.

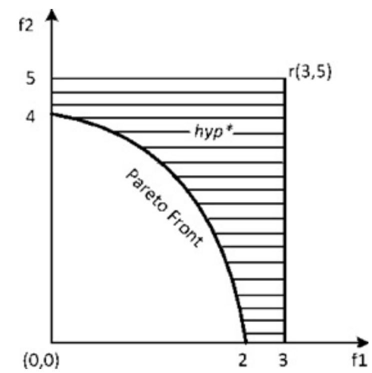


Fig. 17. True hypervolume calculation.

Appendix
True Hypervolume Value of WFG Problems

The true optimal hypervolume values (hyp*) for WFG problems vary by problem, due to the different Pareto-front geometries employed.

From [45], the Pareto-front shape WFG4–9 is part of a hyperellipse with radii $r_i = 2 \times i, i = 1, 2, \ldots, M$. It is a regular geometry. Therefore, the hyp* can be computed by

$$\text{hyp}^* = V_2 - \frac{V_1}{2^M} \tag{7}$$

where V_1 is the volume of M-dimension hyperellipsoid and V_2 is the volume of M-dimension hypercube, which is constructed by the reference point anti-ideal and the coordinate origin. As the volume covered by the Pareto front is only in the first $\frac{1}{2^M}$ area, i.e., the first quadrant for two dimensions. So V_1 must be divided by 2^M. The formula for calculating V_1 is given as follows [63], [64]:

$$V_1 = \begin{cases} \dfrac{1}{(\frac{M}{2})!}\pi^{\frac{M}{2}} \displaystyle\prod_{i=1}^{M} r_i, & M \text{ is even} \\ \dfrac{2^{(\frac{M+1}{2})}}{(M)!!}\pi^{\frac{M-1}{2}} \displaystyle\prod_{i=1}^{M} r_i, & M \text{ is odd.} \end{cases} \tag{8}$$

Taking two-objective WFG4 test as an example, see Fig. 17, $V_1 = \pi r_1 r_2 = \pi \times 2 \times 4 = 8\pi$, and $V_2 = 3 \times 5 = 15$. So hyp$^* = 15 - \frac{8\pi}{2^2} = 8.7168$.

As the Pareto fronts of WFG2 and WFG3 are not regular geometry (WFG2 is disconnected, WFG3 is linear), the above computation process is not available. However, in [45], the authors point out that the optimal solutions of WFG2 and WFG3 satisfy the condition (9)

$$Z_{i=k+1:nvar} = 2 \times i \times 0.35 \tag{9}$$

where $nvar$ is the number of decision variables and $nvar = k + l$, k and l are position and distance parameters. Therefore, we generate $10\,000 \times M$ optimal solutions for M-dimension WFG2 and WFG3 and then the approximation of hyp* can be calculated using conventional methods [22], [47].

REFERENCES

[1] C. Coello Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Computat. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. New York: Wiley, 2001.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Computat.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des. Optimiz. Contr. Applicat. Indust. Problems EUROGEN*, vol. 3242, no. 103. 2002, pp. 95–100.

[5] R. Purshouse and P. Fleming, "Evolutionary many-objective optimisation: An exploratory analysis," in *Proc. IEEE Congr. Evol. Computat.*, vol. 3. Dec. 2003, pp. 2066–2073.

[6] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Computat.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[7] D. Goldberg, *The Race, the Hurdle, and the Sweet Spot*. San Mateo, CA: Morgan Kaufmann, Apr. 1999.

[8] E. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *Proc. CEC*, Sep. 2005, vol. 1, pp. 222–227.

[9] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, May 2009, vol. 1, pp. 1820–1825.

[10] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Computat.*, Jun. 2008, pp. 2424–2431.

[11] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems," in *Proc. 10th Annu. Conf. GECCO*, 2008, pp. 649–656.

[12] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optimiz.*, 2007, pp. 742–756.

[13] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2007, pp. 727–741.

[14] D. Corne and J. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proc. 9th Annu. Conf. Genet. Evol. Computat.*, 2007, pp. 773–780.

[15] S. Kukkonen, S. Member, and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. IEEE Congr. Evol. Computat.*, Sep. 2007, pp. 3983–3990.

[16] H. Sato, H. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2007, pp. 5–20.

[17] J. Branke, "Guidance in evolutionary multi-objective optimization," *Adv. Eng. Softw.*, vol. 32, no. 6, pp. 499–507, Jun. 2001.

[18] H. Ishibuchi and Y. Nojima, "Optimization of scalarizing functions through evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2007, pp. 51–65.

[19] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-A comparative experiment," *IEEE Trans. Evol. Computat.*, vol. 6, no. 4, pp. 402–412, Aug. 2002.

[20] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Computat.*, vol. 10, no. 3, pp. 263–82, Jan. 2002.

[21] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. PPSN VIII*, 2004, pp. 832–842.

[22] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Computat.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[23] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2005, pp. 62–76.

[24] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.

[25] N. Beume, C. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Trans. Evol. Computat.*, vol. 13, no. 5, pp. 1075–1082, Oct. 2009.

[26] J. Bader and E. Zitzler, "A hypervolume-based optimizer for high-dimensional objective spaces," in *Proc. Conf. MOPGP*, vol. 638. 2010, pp. 35–54.

[27] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Computat.*, vol. 19, no. 1, pp. 45–76, Jan. 2011.

[28] Q. Zhang, S. Member, and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Computat.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[29] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. CEC*, no. 1. May 2009, pp. 203–208.

[30] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Computat.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[31] C. Fonseca and P. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Trans. Syst., Man, Cybern, Part A, Syst. Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998.

[32] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Computat.*, vol. 17, no. 3, pp. 411–36, Jan. 2009.

[33] R. Purshouse, C. Jalbă, and P. Fleming, "Preference-driven co-evolutionary algorithms show promise for many-objective optimisation," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2011, pp. 136–150.

[34] K. Deb, M. Mohan, and S. Mishra, "Towards a quick computation of well-spread Pareto-optimal solutions," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2003, pp. 222–236.

[35] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12. New York: Springer, 1999.

[36] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Computat.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[37] Y. Jin and T. Okabe, "Adapting weighted aggregation for multiobjective evolution strategies," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2001, pp. 96–110.

[38] E. Hughes, "Multiple single objective Pareto sampling," in *Proc. IEEE Congr. Evol. Computat.*, Dec. 2003, vol. 4, pp. 2678–2684.

[39] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *Proc. IEEE Congr. Evol. Computat.*, Sep. 2007, pp. 3944–3951.

[40] J. Bongard and H. Lipson, "Nonlinear system identification using coevolution of models and tests," *IEEE Trans. Evol. Computat.*, vol. 9, no. 4, pp. 361–384, Aug. 2005.

[41] M. Kleeman and G. Lamont, "Coevolutionary multi-objective EAs: The next frontier?" in *Proc. CEC*, 2006, pp. 1726–1735.

[42] J. Lohn, W. Kraus, and G. Haith, "Comparing a coevolutionary genetic algorithm for multiobjective optimization," in *Proc. CEC*, May 2002, vol. 2, pp. 1157–1162.

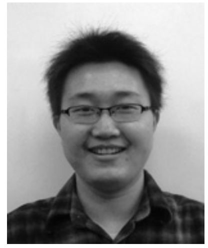
[43] K. Deb, M. Mohan, and S. Mishra, "Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Computat.*, vol. 13, no. 4, pp. 501–25, Jan. 2005.

[44] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Computat.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[45] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Computat.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[46] C. Fonseca and P. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Proc. PPSN VI*, 1996, vol. 1141, pp. 584–593.

[47] C. Fonseca, L. Paquete, and M. López-Ibáñez, "An improved dimension-sweep algorithm for the hypervolume indicator," in *Proc. IEEE CEC*, Jul. 2006, pp. 1157–1163.
[48] M. Hollander and D. Wolfe, *Nonparametric Statistical Methods*. New York: Wiley-Interscience, 1999.
[49] H. Abdi, "The Bonferonni and Šidák corrections for multiple comparisons," *Encyclopedia of Measurement and Statistics*. Thousand Oaks, CA: Sage, 2007.
[50] D. Williamson, R. Parker, and J. Kendrick, "The box plot: A simple visual method to interpret data," *Ann. Internal Med.*, vol. 110, no. 11, p. 916, 1989.
[51] D. Van Veldhuizen and G. Lamont, "On measuring multiobjective evolutionary algorithm performance," in *Proc. IEEE Congr. Evol. Computat.*, Jul. 2000, pp. 204–211.
[52] O. Schütze, A. Lara, and C. A. Coello Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Trans. Evol. Computat.*, vol. 15, no. 4, pp. 444–455, Aug. 2011.
[53] H. Ishibuchi, N. Akedo, H. Ohyanagi, and Y. Nojima, "Behavior of EMO algorithms on many-objective optimization problems with correlated objectives," in *Proc. IEEE Congr. Evol. Computat.*, Jun. 2011, pp. 1465–1472.
[54] H. Ishibuchi, Y. Hitotsuyanagi, H. Ohyanagi, and Y. Nojima, "Effects of the existence of highly correlated objectives on the behavior of MOEA/D," in *Evolutionary Multi-Criterion Optimization*. New York: Springer, 2011, pp. 166–181.
[55] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization," in *Proc. IEEE Congr. Evol. Computat.*, May 2009, pp. 2508–2515.
[56] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. 12th Annu. Conf. Genet. Evol. Computat.*, 2010, pp. 519–526.
[57] E. D. de Jong and J. B. Pollack, "Ideal evaluation from coevolution," *Evol. Computat.*, vol. 12, no. 2, pp. 159–92, Jan. 2004.
[58] D. Hadka and P. Reed. (2011). Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization, *Evol. Computat.* [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/EVCO_a_00053
[59] M. López-Ibánez and T. Stützle, "The automatic design of multi-objective ant colony optimization algorithms," IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, Tech. Rep. TR/IRIDIA/2011-003, Feb. 2011.
[60] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Computat.*, vol. 3, no. 2, pp. 124–141, Jul. 1999.
[61] J. Brest, B. Bošković, S. Greiner, V. Žumer, and M. S. Maučec, "Performance comparison of self-adaptive and adaptive differential evolution algorithms," *Soft Comput.*, vol. 11, no. 7, pp. 617–629, Jul. 2006.
[62] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Computat.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
[63] G. Tee, "Surface area of ellipsoid segment," Dept. Math., School Math. Inform. Sci., Univ. Auckland, Auckland, New Zealand, 2005.
[64] H. Schrocker, "Uniqueness results for minimal enclosing ellipsoids," *Comput.-Aided Geometric Des.*, vol. 25, no. 9, pp. 756–762, Dec. 2008.



Rui Wang was born in Shaanxi, China. He received the B.S. degree in systems engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently pursuing the Ph.D. degree with the Department of Automatic Control and Systems Engineering and Rolls-Royce University Technology Centre, University of Sheffield, Sheffield, U.K.

His current research interests include evolutionary computation and multiobjective optimization.

Robin C. Purshouse received the M.Eng. degree in control systems engineering in 1999 and the Ph.D. degree in control systems, for his research on evolutionary many-objective optimization, in 2004, both from the University of Sheffield, Sheffield, U.K.

He was with Logica plc, Cobham, U.K., from 1999 to 2000, PA Consulting Group, London, U.K., from 2003 to 2007, and Rolls-Royce plc, Derby, U.K., from 2007 to 2008. In 2008, he was a Research Fellow with the School of Health and Related Research, University of Sheffield, where he is currently a Lecturer with the Department of Automatic Control and Systems Engineering.

Peter J. Fleming received the B.S. and Ph.D. degrees from the Queen's University, Belfast, U.K.

Since 1993, he has been the Director with the Department of Control and System Engineering, Rolls-Royce University Technology Centre, University of Sheffield. His control and systems engineering research interests include multicriteria decision making, optimization, grid computing, and industrial applications of modeling, monitoring, and control. He has over 400 research publications, including six books. His current research interests include development of close links with a variety of industries in sectors such as aerospace, power generation, food processing, pharmaceuticals, and manufacturing.

Prof. Fleming is a fellow of the Royal Academy of Engineering, the Institution of Electrical Engineers, and the Institute of Measurement and Control. He is an advisor with the International Federation of Automatic Control. He is the Editor-in-Chief of the *International Journal of Systems Science*.