RAIRO-Oper. Res. 51 (2017) 1301–1315
DOI: 10.1051/ro/2017014

RAIRO Operations Research
www.rairo-ro.org

A NEW HYPERVOLUME-BASED DIFFERENTIAL EVOLUTION ALGORITHM FOR MANY-OBJECTIVE OPTIMIZATION*

Chao Liu[1,2], Qi Zhao[1,2], Bai Yan[3] and Yang Gao[1,2]

Abstract. Evolutionary algorithms are successfully used for many-objective optimization. However, solutions are prone to become nondominated from each other with the increase in the number of objectives, which reduces the efficiency of Pareto dominance-based algorithms. In this paper, a new hypervolume-based differential evolution algorithm (MODEhv) is proposed for many-objective optimization problems (MaOPs). In MODEhv, a modified differential evolution paradigm with automatic parameter configuration strategy is used to balance exploration and exploitation of the algorithm. Besides, the hypervolume indicator is incorporated for the selection of solutions to be varied and solutions to be kept in archive respectively. Finally, a threshold technique is employed to improve diversity of solutions in archive. MODEhv is investigated on a set of widely used benchmark problems and compared with five state-of-the-art algorithms. The experimental results show the efficiency of MODEhv for solving MaOPs.

Mathematics Subject Classification. 65K10,68T20.

Received February 17, 2016. Accepted March 2, 2017.

1. Introduction

Many real-world applications can be modeled as multi-objective optimization problems (MOPs) which require optimizing several conflicting objectives simultaneously [15]. According to whether the decision variables of problems are continuous or discrete, MOPs can be roughly placed in continuous problems and discrete (combinational) problems [21]. Multi-objective evolutionary algorithms (MOEAs) have been successfully used in both types of MOPs [15, 33], mainly due to the following properties: (i) They search for multiple solutions at a time whereas conventional techniques may only obtain one solution in a single run. (ii) They are designed as effective and robust optimizers without any prior knowledge. (iii) They are less influenced by the geometries of the Pareto front (PF) of MOPs than mathematical programming approaches are.

MOEAs have been widely used in solving problems with two or three objectives [15, 33]. However, there are many real-world problems that have more than four objectives (the so called many-objective optimization problems, MaOPs), especially in continuous problems domain, such as water distribution [12] car controller optimization [17] and land use management [7]. These MaOPs pose great difficulties to conventional MOEAs.

Keywords. Differential Evolution, Hypervolume indicator, Many-objective optimization, Many-objective evolutionary algorithm.

* *This paper is supported by the National Natural Science Foundation of China under Grant 61273230, 61603011 and 61603010, Shandong Financial Collaborative Innovation Center under Grant 14AWTJ01-4, Capital Social development and Social Management Collaborative Innovation Center, and Research Base of Beijing Modern Manufacturing Development.*

[1] College of Economics and Management, Beijing University of Technology, Beijing 100124, China.

[2] Research Base of Beijing Modern Manufacturing Development, Beijing 100124, China. qzhao@emails.bjut.edu.cn

[3] Institute of Laser Engineering, Beijing University of Technology, Beijing 100124, China. yanbai@emails.bjut.edu.cn

Article published by EDP Sciences

Firstly, the Pareto dominance-based selection operator commonly used in MOEAs is inefficient in MaOPs. The reason is that solutions are prone to become nondominated with each other with the increase in the number of objectives, thus the selection pressure towards the true PF gets weak, and the search ability of the algorithm decreases [11, 19, 25]. In addition, variation operator may lose efficacy in MaOPs, as solutions tend to be distant from each other in high-dimensional space, thus the offspring may also be distant from their parents [6, 24].

To address the difficulties with conventional MOEAs in solving continuous MaOPs, a category of MOEAs makes use of indicator-based methods. The most prevalent indicator is the hypervolume indicator introduced by Zitzler [31]. A unique feature of the hypervolume indictor is that it is the only indicator which has a strictly monotonic relation with Pareto dominance [32]. This merit is of great value in solving MaOPs since Pareto dominance loses efficacy in such scenario [11]. To date, the hypervolume indicator was mostly used as a secondary criterion to rank a particular front after Pareto dominance sorting [13, 14]. Bader and Zitzler [3] proposed a hypervolume-based fitness assignment method and provided an algorithm HypE which tailored specifically to the hypervolume indicator. However, to the best of our knowledge, there are still lack of effective many-objective optimization algorithms combing the merits of the hypervolume indicator with other proper operators.

For the variation operators of MOEAs, genetic evolution is commonly used in well-known MOEAs such as NSGA-II [8] and SPEA2 [34]. In recent years, new nature-inspired metaheuristics including ant colony evolution (ACO), particle swarm optimization (PSO), differential evolution (DE) and some other paradigms are introduced in MOEAs [33]. Among the existing metaheuristics, DE is arguably one of the most robust and simplest algorithms [9]. Abbass *et al.* [2] firstly employed DE to solve continuous MOPs, and proposed an algorithm called Pareto DE (PDE). Kukkonen and Lampinen [26] developed GDE3 which is an enhanced version of GDE [20], experiments verified that GDE3 was effective for solving MOPs. Later, Li and Zhang [16] proposed an algorithm MOEA/D-DE which is based on decomposition to solve MOPs with complicated geometries of Pareto fronts. An algorithm MODEA was developed by Ali *et al.* [1] for bi-objective and tri-objective problems. The algorithm incorporated opposition based learning in population initialization and a local search strategy in variation, experiments showed its effectiveness. To date, DE has been successfully used in MOPs, it is valuable to introduce such powerful paradigm in solving MaOPs [9].

This paper proposes a novel algorithm MODEhv for solving continuous MaOPs. In MODEhv, care is taken in three main parts. (i) A modified DE with automatic parameter configuration strategy is used to guide mutation and crossover processes, which is expected to balance exploration and exploitation of the algorithm. (ii) The hypervolume indicator is applied as selection operator for the selection of solutions to be varied and to be kept in memory respectively, for sake of avoiding low efficiency of Pareto dominance in MaOPs. (iii) A threshold-based diversity maintenance technique is used in updating archive, which aims to avoid the higher ranked solution filling up the entire population of the archive.

The reminder of the paper is as follows. Section 2 describes definitions related to the research. Section 3 provides the proposed MODEhv algorithm in detail. Thereafter, experimental setup and performance metrics are given in Section 4. Experimental results and discussion are given in Section 5. Finally, Section 6 concludes the article.

2. Background

The terminologies, formulations and concepts related to this paper are introduced briefly in this section.

2.1. Many-objective optimization problems

Problems with a number of conflicting objectives can be formulated as follows:

$$\begin{aligned} \min \ & F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ \text{subject to} \ & g_i(x) \leq 0, i = 1, 2, \ldots, q \\ & h_j(x) = 0, j = 1, 2m \ldots p \\ & X \in \Omega \ , \end{aligned} \tag{2.1}$$

where $x \in \Omega$ is the decision space, $x = (x_1, x_2, \ldots, x_n)$is a andidate solution. $F : \Omega \rightarrow \mathbb{R}^m$ constitutes m objective functions to be minimized, and $\mathbb{R}^m$ is defined as the objective space. $g_i(x)$ and $h_j(x)$ are the inequality and equality constraints respectively that the problem should satisfy. The aim of solving MaOPs is to obtain these candidate solutions X that optimize the objective functions as well as satisfy the constraints.

2.2. Pareto dominance

Considering a minimization MOP, for two candidate solutions x_1 and $x_2 \in \Omega$, if the conditions below are both satisfied:

$$1.\forall i \in 1, 2, \ldots, m : f_i(x_1) \leq f_i(x_2)$$

$$2.\exists j \in 1, 2, \ldots, m : f_j(x_1) < f_j(x_2),$$

then x_1 dominates x_2 (or $x_1 \prec x_2$). Alternatively, x_1 dominates x_2 if and only if x_1 is not worse in every objectives, and better in at least one objective than x_2. Solution x_1 and x_2 are called incomparable (or $x_1 \parallel x_2$) if and only if neither x_1 nor x_2 could dominate the other one, that is, x_1 and x_2 are non-dominated from each other.

2.3. Differential evolution

Differential Evolution is a simple and robust evolutionary algorithm proposed by Storn and Price [23]. It starts with an initial population P containing N solutions of a MOP and uses mutation, crossover and selection operators to update the population.

Mutation: For a target vector $x_{i,G} = (x_{1,i,G}, x_{2,i,G}, \ldots, x_{j,i,G})$ where $x_{i,G}$ is the ith solution of P in generation G and $x_{j,i,G}$ is the jth decision variable of $x_{i,G}$, its donor vector $v_{i,G}$ is generated as follows:

$$v_{i,G} = x_{r1,G} + F \times (x_{r2,G} - x_{r3,G}) \tag{2.2}$$

where $x_{r1,G}$, $x_{r2,G}$,$x_{r3,G}$ are three distinct vectors randomly selected from the population and different from $x_{i,G}$. $F \in (0, 1)$ is a scaling factor controlling the disturbance towards $x_{r1,G}$.

Crossover: for $x_{i,G}$ and its donor vector $v_{i,G}$, a crossover operator is used to generate the trial vector $u_{i,G}$, as follows:

$$u_{i,G} = \begin{cases} v_{j,i,G} & \text{if rand(0,1)} \leq \text{Cr or } j = \text{rand}\{1, 2, \ldots, n\}. \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{2.3}$$

where n is the number of decision variables, and $Cr \in (0, 1)$ is the crossover ratio controlling the crossover probability of the decision variables of the vectors.

DE algorithms with larger F and Cr values have higher ability in exploration (*i.e.* search larger regions in the objective space). By contrast, relatively small F and Cr values result in higher efficiency in exploitation (*i.e.* local search within the neighborhood of current solutions). However, if the algorithm holds both small F and Cr values during the evolution process, the generated population may lose diversity and fail into premature convergence [28]. Therefore, F and Cr should be carefully designed according to the optimization problems so as to balance exploration and exploitation of DE algorithms [28].

Selection: for a target vector $x_{i,G}$ and its trail vector $u_{i,G}$, a selection operator is used to choose the better one and send it to the next generation, as follows:

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{if} \quad u_{i,G} \prec x_{i,G} \\ x_{i,G} & \text{otherwise} \end{cases} \tag{2.4}$$

The process above will be executed for N times to generate and select N solutions for the next generation. The DE model above is termed as $DE/rand/1/bin$, more versions of DE can be found from study [9].

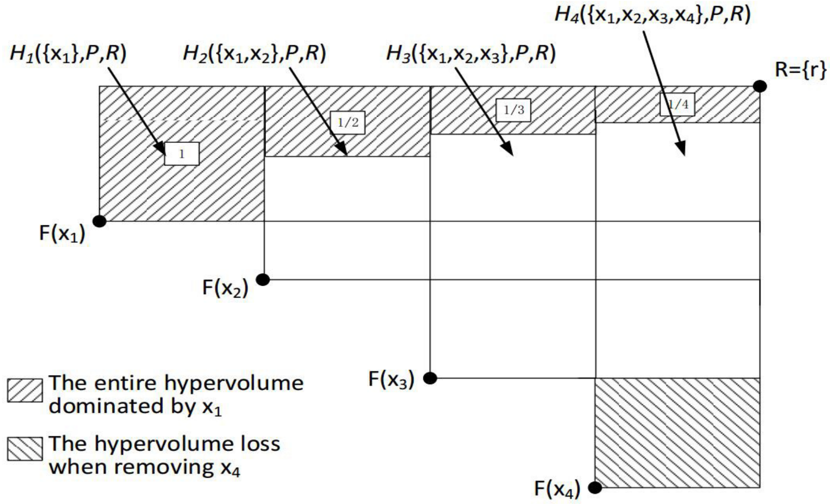


FIGURE 1. The entire hypervolume dominated by solution x_1 and the hypervolume loss when removing x_4.

2.4. Hypervolume Indicator

Hypervolume indicator is given by Zitzler [31]. For a solution set $P \in \Omega$ of a MOP and a reference set $R \in Z$, the hypervolume value of P in objective space can be calculated as the Lebesgue measure of the union of hypercubes between P and R, as follows [29]:

$$I_H(P, R) := \lambda(H(P, R)) \tag{2.5}$$

where λ is the Lebesgue measure,$H(P, R)$ is the union of hypercubes between P and R.

Entire hypervolume dominated by a solution: for a solution set $P = \{x_1, x_2, x_3, x_4\}$ and a reference set $R = \{r\}$, $H(x_1, P, R)$ is the entire hypervolume dominated by x_1 in objective space (as shown in Fig. 1). The hypervolume of each subspace $H(X, P, R)$ can be distributed equally among the dominating solutions $x \in X$ [3]. For example, in Figure 1, x_1 and x_2 are distributed half of $H_2(\{x_1, x_2\}, P, R)$ respectively. According to the principles above, the hypervolume value of a solution $x \in P$ can be calculated as follows [3]:

$$I_h(x, P, R) := \sum_{i=1}^{|P|} \frac{1}{i} \lambda(H_i(x, P, R)) \tag{2.6}$$

It has been proved that solution x_1 dominates x_2 (or $x_1 \prec x_2$), as long as $I_h(x_1, P, R) > I_h(x_2, P, R)$, and x_1 also has a better diversity performance than x_2 [32], which means that a solution with a bigger hypervolume value is better in both convergence and diversity.

Hypervolume loss: given a solution set $P = \{x_1, x_2, x_3, x_4\}$ and a reference set $R = \{r\}$, according to [3], the value of hypervolume loss of P when removing a solution $x \in P$ can be calculated by extending equation (2.6), as follows:

$$I_h^k(x, P, R) := \sum_{i=1}^{k} \frac{a_i}{i} \lambda(H_i(x, P, R)) \tag{2.7}$$

where $a_i := \prod_{j=1}^{i-1} \frac{1-j}{|P|-j}$, k is the number of solutions needed to be removed. It has been demonstrated that a solution set with a bigger hypervolume value strictly has a better performance in both convergence and diversity [32], therefore, if a solution needs to be removed from a population, the solution cutting off which will lead a minimum hypervolume loss of the population could be removed.

3. Proposed algorithm

In this section, the proposed MODEhv is described in detail. MODEhv starts by uniform randomly generating an initial population P_0 containing N initial solutions, in each generation G, the variation operator is implemented in P_G to generate an offspring solution set Q_G, thereafter a selection operator is employed to select N solutions from $P_G \bigcup Q_G$ as P_{G+1}, the above process will iterate until $G = G_{max}$. Each component of MODEhv is introduced as follows.

3.1. Variation

In original DE ($DE/rand/1/bin$) shown in Section 2.3, three parameter solutions $x_{r1,G}, x_{r2,G}, x_{r3,G}$ are selected randomly from the population and the base solution $x_{r1,G}$ is chosen arbitrarily from the three. This completely random strategy is effective in maintaining diversity of solutions, but leads to a slow speed of convergence towards the true PF. An alternative strategy commonly used is $DE/best/1/bin$, which always chooses the best solution of the current population as the base solution. This greedy strategy has a better convergence performance in early generations, but it may lose diversity of the solutions and lead to premature convergence with the evolution progresses [1].

In MODEhv, a modified *DE/best/1/bin* strategy with randomized localization is proposed to balance the capabilities of convergence and diversity maintenance. First, three parameter solutions $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ are selected randomly from the population P; then, the solution with the highest fitness among the three is chosen as the base solution x_{best}. This strategy retains the random selection of three parameter solutions to maintain diversity, meanwhile provides a property of local search in different regions of spaces around the parameter solutions. Assuming $x_{r1} = x_{best}$, the modified strategy is implemented as follows:

$$v_i = x_{best} + F \times (x_{r2} - x_{r3}) \tag{3.1}$$

Fitness assignment of DE parameter solutions: in most existing MOEAs, Pareto dominance is used in fitness assignment [15], however, it loses efficiency in MaOPs [25]. Here, a hypervolume-based fitness assignment method is used for assigning fitness to parameter solutions. According to the properties of the entire hypervolume dominated by a solution shown in Section 2.4, the fitness of DE parameter solutions could be determined equal to their entire dominating hypervolume values respectively. In other words, the solution with the biggest entire dominating hypervolume value among $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ has the highest fitness and will be selected as x_{best}. The entire dominating hypervolume value is calculated following equation (2.6), in which Monte Carlo simulation is used [3] to reduce the high computation effort.

After that, a binomial crossover is used to improve the diversity of solutions following equation (2.3).

Automatic DE parameter configuration strategy: evolutionary algorithms should well balance exploration and exploitation so as to obtain solutions with good performance of convergence and diversity. However, there is not a specific configuration of algorithm that suits for different problems [23]. To improve the applicability of MODEhv in various problems, an automatic DE parameter configuration strategy is developed, which is inspired by ensemble learning [27], that is, solutions are varied following different parameter configurations, the fittest configuration will be implemented in more solutions in the next generation. The proposed strategy is as follows.

Step 1. Three F and Cr randomly generating schemes are designed:

$$\textit{Scheme } 1 \ F = \text{rand}\{0.1, 0.2, 0.3\}, Cr = \text{rand}\{0.1, 0.2, 0.3\}$$

$$\textit{Scheme } 2 \ F = \text{rand}\{0.4, 0.5, 0.6\}, Cr = \text{rand}\{0.4, 0.5, 0.6\}$$

$$Scheme\ 3\ F = \mathrm{rand}\{0.7, 0.8, 0.9\}, Cr = \mathrm{rand}\{0.7, 0.8, 0.9\}$$

where *Scheme* 1 emphasizes on exploitation, *Scheme* 3 focuses on exploration, while *Scheme* 2 considers the balances.

Step 2. The current population P_G is divided into 4 subpopulations $P_G^j,\ j = 1, 2, \ldots, 4$ with size of $\alpha_j * N$ respectively, and satisfies $\alpha_1 = \alpha_2 = \alpha_3 \ll \alpha_4$.

Step 3. *Scheme* 1, 2, 3 are implemented in P_G^1, P_G^2 and P_G^3 respectively. *Scheme* 4 is the same as the scheme with the highest fitness among *Scheme* $k,\ k = 1, 2, 3$ in generation $G - 1$. It is implemented in P_G^4. The fitness of *Scheme* k in generation $G - 1$ is calculated as $|Q_{G-1}^j \cap P_G|$, where $j = k$, Q_{G-1}^j is the set of offspring solutions generated by P_{G-1}^j with *Scheme k*. *Step 4*, $\alpha_j * N$ combinations of F and Cr are configured following the corresponding scheme. Specifically, in generation $G = 1$, the fitness of the three schemes are determined in equal. If there are more than one fittest scheme, solutions in P_G^4 are randomly implemented on the fittest schemes respectively. The corresponding pseud-code is shown in line 4-13 of Algorithm 1.

Following the above strategy, the parameter F and Cr in MODEhv could be automatically configured to balance exploration and exploitation in different stages of evolution dynamically when solving different problems.

3.2. Selection

In original DE shown in Section 2.3, a target vector and its trail vector are compared in pairs, and the better one is selected according to their Pareto dominance relation. This process is executed for N times to choose N solutions for the next generation. This greedy strategy performs inefficiently in maintaining diversity and may lead to local convergence [1]. In MODEhv, the N target solutions (parents) and the N trail solutions (offspring) are combined in a temporary population P_{temp} with a size of $2N$, then N better solutions are selected from P_{temp} by the selection operator. This strategy provides an effect in global optimal checking among the parent and offspring solutions.

Based on the above strategy, a hypervolume loss-based selection operator is incorporated in MODEhv to truncate the $2N$ solutions to N.

Step 1. the Non-Dominated Sorting method borrowed from NSGA-II [8] is used to sort the $2N$ solutions into several non-domination levels $R_1, R_2, \ldots, R_l$.

Step 2. starting with the highest ranked solutions, the solutions are moved to the archive P_{G+1} one by one, until P_{G+1} reaches N size.

Step 3. assuming R_q is the last included level of P_{G+1}, if the current size of P_{G+1} is larger than N, the hypervolume loss-based selection is used to sort the solutions in R_q. In detail, the hypervolume loss with respect to each solutions in current R_q is calculated by equation (2.7), then the solution with the minimum hypervolume loss is eliminated from the current R_q. This process is executed recursively until the size of P_{G+1} decreases to N. Its pseud-code is shown in line 22-31 of Algorithm 1.

An additional threshold-based diversity maintenance technique is used in Step 2. It has been verified that the number of solutions in R_1 may be more than N in MaOPs [24]. If the new population is full of solutions from R_1, the algorithm may get stuck to a local optima [4]. In MODEhv, if the count of solutions in R_1 is between $\beta * N$ and $N + *N$, only $\beta * N$ solutions in R_1 could be access to Step 2, which aims to maintain the diversity of solutions. The β percentage of N solutions are selected by the hypervolume loss-based selection operator concerning the solutions in R_1. The above process is shown in line 17-21 of Algorithm 1.

According to each component of MODEhv introduced above, the pseud-code of the algorithm is shown in Algorithm 1.

Algorithm 1: Pseud-code of MODEhv

Input: subpopulation size control parameter α_1 (as $\alpha_1 = \alpha_2 = \alpha_3$, $\alpha_4 = 1 - 3 * \alpha_1$); diversity maintenance threshold β
Output: P
$P_0 \leftarrow Population\ Initialization$;
$G = 1$, $P_G \leftarrow P_0$, $Q_G \leftarrow \emptyset$;
while $G \leq G_{max}$ **do**
    /*Variation*/;
    $Scheme\ 4 \leftarrow the\ Scheme\ with\ the\ highest\ fitness\ among\ 1, 2, 3\ in\ G - 1$;
    for $i = 1 : N$ **do**
        for $j = 1 : 4$ **do**
            $x_i^j \leftarrow assign\ x_i to\ P_G^j until\ |P_G^j|\ reaches\ \alpha_j * N$;
            $F_i^j, Cr_i^j \leftarrow configure\ Fand\ Crfor\ x_i^j\ following\ Scheme\ j$;
            $u_i \leftarrow Mutation(F_i^j) + Crossover(Cr_i^j)$;
            $Q_G \leftarrow Q_G \cup u_i$;
        end
    end
    /*Selection*/;
    $P_temp \leftarrow P_G \cup Q_G$;
    $divide\ P_{\text{temp}}\ into\ nondomination\ levels\ R_1, R_2, \ldots, R_l\ by\ Nondominated\ Sorting$;
    $k = |R_1| - \beta * N$;
    while $0 < k \leq N$ **do**
        $x_{minHVloss} \leftarrow\ solution\ with\ the\ min.\ hypervolume\ loss\ from\ R_1$;
        $R_1 \leftarrow R_1 \setminus x_{minHVloss}, k = k - 1$;
    end
    $q = 1$;
    while $|R_1| + |R_2| + \ldots + |R_q| < N$ **do**
        $q = q + 1$;
        $P_{G+1} \leftarrow R_1 \cup R_2 \cup \ldots \cup R_q$;
    end
    $k = |R_{G+1}| - N$;
    while $k > 0$ **do**
        $x_{minHVloss} \leftarrow\ solution\ with\ the\ min.\ hypervolume\ loss\ from\ R_q$;
        $R_q \leftarrow R_q \setminus X_{minHVloss}, k = k - 1$;
    end
    $update\ P_{G+1}, P_{\text{temp}} \leftarrow \emptyset$;
    $G = G + 1$;
end
return P;

4. Experiment

4.1. Experimental Setup

The proposed MODEhv is compared with five state-of-the-art MOEAs: NSGA-II [8], IBEA [30], SMPSO [18], IBMOLS [5] and HypE [3]. For MODEhv, α_1 and $s\beta$ are taken as 0.15 and 0.75 respectively, the configuration of α_1 and β is justified in Section 5.4, and 10 000 samples are set per simulation in Monte Carlo simulation. The parameters of the comparing algorithms are set according to the recommendations in [3,5,8,18,30] respectively.

The performance of algorithms are validated on DTLZ1 to DTLZ7 of the DTLZ test suite [10], each with five to twenty-five objectives. As per recommendation [10], the number of decision variables $n = m + k - 1$, where m is the number of objectives, $k = 5$ for DTLZ1 and $k = 10$ for DTLZ2 to DTLZ7 problems. For all benchmark problems, each algorithm runs 20 times, and both the population size and the maximum number of generation are set to 100 for fair comparison.

4.2. Performance metrics

A. Hypervolume. Hypervolume (HV) [31] is a valid quality indicator which can measure both convergence and diversity of solutions towards the true PF (details of HV is shown in Sect. 2.4). For two solution sets A and B and a predefined reference point $R = r$, A has a better performance both in convergence and diversity than B if $I_H(A, R) > I_H(B, R)$. The HV values in this study are normalized to the minimal and maximal values obtained per instance. Monte Carlo simulation is used to calculate the HV values [3], and 100 000 samples are used per simulation.

B. Wilcoxon sign test. Wilcoxon sign [22] is a nonparametric test for differences between two paired (or matched) samples. The Wilcoxon sign statistics here tests results at a 0.05 significance level. Specially, if the statistical test does not pass between two algorithms, both of them are determined to rank at the same level. For example, given two solution sets A and B, if A is not better than B and B is also not better than A both at a 0.05 significance level, then A and B are ranked at the same level.

C. Performance score. In this study, the performance score P is proposed to show the performance of algorithms when considering a specific test instance. For a selected algorithm A, its performance score $P(A)$ is defined as the number of algorithms which are better than A with respect to the Wilcoxon sign test of the HV values at a 0.05 significance level. P can be calculated as follows:

$$P(A_i) = \sum_{j=1(j \neq i)}^{l} \beta_{i,j}, \tag{4.1}$$

where i and j are serial numbers of algorithms, l is the amount of participant algorithms. If A_j is better than A_i, then $\beta_{i,j} = 1$, otherwise $\beta_{i,j} = 0$. Algorithm with smaller P value has a better performance with respect to the HV value.

5. Experimental results and discussion

A summary of the experimental results are shown in Table 1 and Figure 2. Table 1 gives the average HV values and standard deviation of HV values of different algorithms in per instance, the numbers in brackets are the performance scores of the selected algorithms. Figure 2 shows the HV values of (a) NSGA-II, (b) IBEA, (c) SMPSO, (d) IBMOLS, (e) HypE and (f) MODEhv in benchmark problems with five, seven, ten, and twenty-five objectives respectively. From the experimental results, MODEhv wins 124 out of 140 comparisons, thus it can be claimed that MODEhv is an efficient optimizer for continuous MaOPs.

5.1. Performance comparison with respect to the dimension of problems

According to Table 1, in problems with five objectives, MODEhv obtains the best performance in three problems and the third-best performance in another three problems, which means that MODEhv is competitive in five-objective instances. In problems with seven, ten and twenty-five objectives, MODEhv performs best in five, six and six of seven problems respectively, which clearly indicates its effectiveness in high-dimension instances.

TABLE 1. HV values and performance scores of different algorithms.

Problem	NSGA-II	IBEA	SMPSO	IBMOLS	HypE	MODEhv
			Five objectives			
DTLZ1	(4)0.4698 0.2828	(2)0.7793 0.0782	(4)0.4440 0.1812	(2)0.7809 0.0508	(1)0.8127 0.0511	(0) 0.9914 0.0168
DTLZ2	(4)0.3630 0.2000	(0)0.9969 0.0009	(5)0.2919 0.1025	(0)0.9969 0.0008	(2)0.9918 0.0025	(0) 0.9937 0.0063
DTLZ3	(4)0.3744 0.1983	(1)0.8586 0.0075	(4)0.3557 0.1440	(1)0.8594 0.0009	(0)0.8612 0.0159	(0) 0.8722 0.0503
DTLZ4	(4)0.2727 0.1312	(0)0.9980 0.0010	(4)0.2573 0.0691	(0)0.9974 0.0027	(3)0.9581 0.0393	(2) 0.9942 0.0046
DTLZ5	(4)0.1805 0.1149	(0)0.9277 0.0347	(5)0.1623 0.1011	(0)0.9172 0.0568	(2)0.6447 0.0566	(3) 0.4713 0.0560
DTLZ6	(5)0.0605 0.0588	(0)0.9978 0.0015	(4)0.2511 0.0292	(0)0.9980 0.0018	(3)0.7446 0.2054	(2) 0.8856 0.0164
DTLZ7	(5)0.4317 0.0548	(1)0.7238 0.0153	(4)0.3309 0.0929	(0)0.7909 0.0926	(2)0.4068 0.2436	(2) 0.4173 0.0759
			Seven objectives			
DTLZ1	(3)0.8704 0.0243	(2)0.9078 0.0067	(5)0.5059 0.1754	(4)0.8534 0.0770	(1)0.9159 0.0043	(0) 0.9590 0.0353
DTLZ2	(4)0.3506 0.1836	(0)0.9968 0.0003	(5)0.2130 0.1970	(3)0.9810 0.0069	(1)0.9937 0.0018	(0) 0.9943 0.0027
DTLZ3	(4)0.5364 0.1178	(1)0.9521 0.0054	(5)0.3919 0.2798	(3)0.8208 0.1346	(0)0.9537 0.0096	(0) 0.9611 0.0310
DTLZ4	(4)0.3080 0.1936	(0)0.9996 0.0003	(5)0.2039 0.1141	(2)0.9702 0.0071	(2)0.9681 0.0016	(1) 0.9798 0.0097
DTLZ5	(4)0.2677 0.1002	(0)0.9482 0.0635	(5)0.1921 0.1606	(2)0.9063 0.0577	(3)0.8225 0.0380	(0) 0.9428 0.0225
DTLZ6	(4)0.1692 0.0399	(1)0.9809 0.0068	(4)0.1599 0.0277	(2)0.9657 0.0105	(3)0.5299 0.3146	(0) 0.9840 0.0162
DTLZ7	(2)0.5956 0.1929	(0)0.9935 0.0040	(3)0.5003 0.1468	(0)0.9881 0.0071	(4)0.3418 0.2222	(3) 0.4965 0.1750
			Ten objectives			
DTLZ1	(1)0.8127 0.0122	(3)0.7927 0.0109	(5)0.2534 0.1653	(4)0.5900 0.1294	(1)0.8175 0.0108	(0) 0.8560 0.0051
DTLZ2	(4)0.2671 0.1590	(1)0.9735 0.0020	(5)0.1710 0.1542	(3)0.7470 0.0295	(1)0.9642 0.0103	(0) 0.9806 0.0127
DTLZ3	(4)0.3812 0.2068	(1)0.9693 0.0092	(5)0.2113 0.2364	(3)0.9225 0.0772	(0)0.9734 0.0150	(0) 0.9786 0.0112
DTLZ4	(4)0.4132 0.2100	(0)0.9959 0.0004	(5)0.2302 0.1647	(3)0.9539 0.0050	(2)0.9796 0.0173	(0) 0.9948 0.0037
DTLZ5	(4)0.2513 0.1482	(1)0.8901 0.0510	(5)0.2141 0.1549	(3)0.7493 0.0254	(2)0.8748 0.0281	(0) 0.8913 0.0467
DTLZ6	(4)0.2915 0.0640	(1)0.9838 0.0104	(5)0.2488 0.0790	(2)0.7707 0.0112	(3)0.5143 0.3218	(0) 0.9900 0.0060
DTLZ7	(4)0.0950 0.0673	(0)0.9869 0.0157	(5)0.0818 0.0490	(1)0.8016 0.0102	(3)0.1552 0.0594	(2) 0.5059 0.0088
			Twenty- five objectives			
DTLZ1	(2)0.7079 0.1030	(3)0.4781 0.1353	(4)0.4112 0.2301	(5)0.3076 0.0960	(1)0.8433 0.0705	(0) 0.8625 0.1442
DTLZ2	(4)0.2540 0.1543	(1)0.9842 0.0042	(5)0.1833 0.1467	(3)0.9082 0.0171	(2)0.9323 0.0589	(0) 0.9892 0.0130
DTLZ3	(4)0.4060 0.2216	(2)0.9817 0.0069	(5)0.3184 0.2105	(3)0.8093 0.0315	(0)0.9911 0.0075	(0) 0.9944 0.0044
DTLZ4	(4)0.6145 0.1719	(0)0.9839 0.0021	(5)0.3982 0.2749	(3)0.7562 0.0452	(2)0.9719 0.0098	(0) 0.9839 0.0297
DTLZ5	(4)0.4806 0.1938	(2)0.9801 0.0155	(5)0.3728 0.1956	(3)0.7944 0.0168	(0)0.9821 0.0064	(0) 0.9844 0.0056
DTLZ6	(4)0.2326 0.0247	(1)0.9672 0.0312	(5)0.1619 0.1094	(3)0.7308 0.0318	(2)0.7406 0.1085	(0) 0.9726 0.0165
DTLZ7	(5)3.4e-05 8.7e-05	(0)0.7884 0.1326	(3)0.3144 0.1005	(4)0.5524 0.0690	(2)0.1289 0.0603	(1) 0.5577 0.1528

5.2. Performance comparison with respect to specific test instances

DTLZ1 and DTLZ3 are test problems with $11^k - 1$ and $3^k - 1$ local optimal fronts in objective space respectively, which makes them difficult to converge to the global true PFs. According to the Table 1 and Figure 2, MODEhv consistently obtains the best performance in all test scenarios of DTLZ1 and DTLZ3, briefly indicating its promise in achieving global optima. HypE, a hypervolume-based algorithm as well, wins the second-best in DTLZ1 and is comparable with MODEhv in DTLZ3 with five, seven and ten objectives. In contrast, NSGA-II and SMPSO make poor performance in most cases.

DTLZ2 is a general convex problem, this type of problems frequently appears in practice. MODEhv makes the best performance in all instances. IBEA also shows a well performance in most cases. IBMOLS, a local search based algorithm, is comparable with MODEhv in five-objective scenario. However, it loses efficiency in high-dimension instances, because local search operators may not perform well in exploration with the increase in the dimension of objective space. Figure 3 shows the non-dominated solutions generated by each algorithm in ten-objective scenario. It can be seen that solutions obtained by MODEhv and HypE are similar in both convergence and diversity. The diversity of solutions generated by IBEA and IBMOLS are worse than those of MODEhv and HypE, while NSGAII and SMPSO could be eligible in neither convergence nor diversity compared with other algorithms.

DTLZ4 is used to measure the capacity of maintaining diversity of solutions. MODEhv performs the third-best and the second-best in problem with five and seven objectives respectively, and reaches the best in ten and twenty-five-objective instances. These results indicate its competitiveness in maintaining diversity of solutions compared with the state-of-the-art algorithms. IBEA ranks first in all instances, showing its ability to maintain diversity.

DTLZ5 is applied to investigate the ability to converge to a curve. MODEhv ranks fourth in problems with five objectives and reaches the first in other instances. These results indicate its applicability in high-dimension

TABLE 2. Comparison of MODEhv to HypE with respect to the HV value.

Problem	HypE	MODEhv	HypE	MODEhv
	Five objectives		Seven objectives	
DTLZ1	0.8127 0.0511	0.9914 0.0168	0.9159 0.0043	0.9590 0.0353
DTLZ2	0.9918 0.0025	0.9937 0.0063	0.9937 0.0018	0.9943 0.0027
DTLZ3	0.8612 0.0159	0.8722 0.0503	0.9537 0.0096	0.9611 0.0310
DTLZ4	0.9581 0.0393	0.9942 0.0046	0.9681 0.0016	0.9798 0.0097
DTLZ5	0.6447 0.0566	0.4713 0.0560	0.8225 0.0380	0.9428 0.0225
DTLZ6	0.7446 0.2054	0.8856 0.0164	0.5299 0.3146	0.9840 0.0162
DTLZ7	0.4068 0.2436	0.4173 0.0759	0.3418 0.2222	0.4965 0.1750
	Ten objectives		Twenty-five objectives	
DTLZ1	0.8175 0.0108	0.8560 0.0051	0.8433 0.0705	0.8625 0.1442
DTLZ2	0.9642 0.0103	0.9806 0.0127	0.9323 0.0589	0.9892 0.0130
DTLZ3	0.9734 0.0150	0.9786 0.0112	0.9911 0.0075	0.9944 0.0044
DTLZ4	0.9796 0.0173	0.9948 0.0037	0.9719 0.0098	0.9839 0.0297
DTLZ5	0.8748 0.0281	0.8913 0.0467	0.9821 0.0064	0.9844 0.0056
DTLZ6	0.5143 0.3218	0.9900 0.0060	0.7406 0.1085	0.9726 0.0165
DTLZ7	0.1552 0.0594	0.5059 0.0088	0.1289 0.0603	0.5577 0.1528

scenarios. IBEA also obtains good results in low-dimension instances. DTLZ6 is a similar but more complex version compared with DTLZ5. MODEhv obtains the best value in cases with seven to twenty-five objectives, following by IBEA, IBMOLS and HypE. In contract, NSGA-II and SMPSO perform badly in all instances of DTLZ5 and DTLZ6.

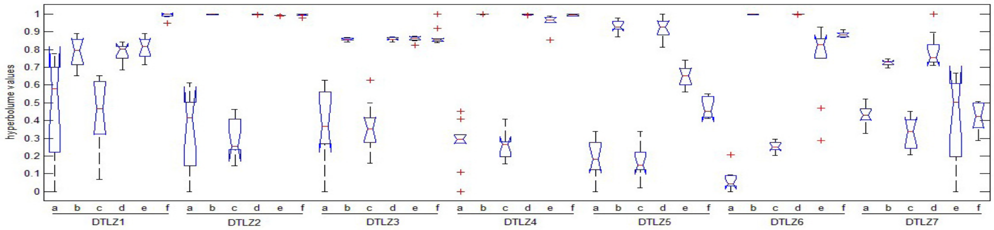
DTLZ7 is a discontinuous problem, MODEhv ranks second or third in five-, ten- and twenty-five- objective instances and ranks fourth in seven-objective case. These results indicate that the proposed algorithm dose not perform well in discontinuous problems, but the value it reached are acceptable in comparison with other algorithms. Thus, MODEhv is not outstanding but eligible in coping with discontinuous problems.

5.3. Performance comparison with HypE

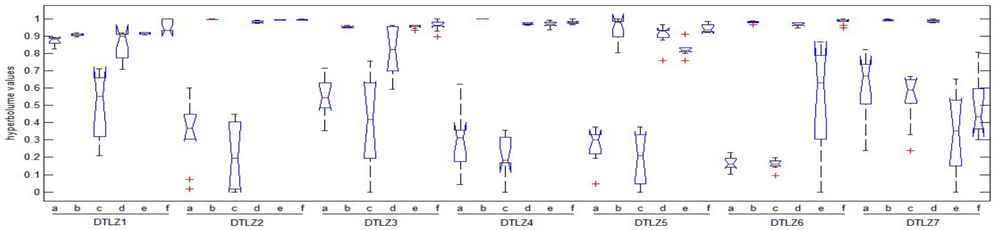
Both MODEhv and HypE employ the hypervolume indicator as selection operator, but they differ in variation and other specific strategies. HypE employs a polynomial distribution and the SBX-20 operator, while MODEhv uses a modified DE paradigm and additional techniques (such as the threshold-based diversity maintenance) tailored specifically to MaOPs. The performance of MODEhv and HypE are compared to investigate the efficiency of the modified DE paradigm and the additional techniques in MODEhv.

A summary of the comparison of MODEhv to HypE are shown in Table 2 and Figure 4. Table 2 shows the average HV values and standard deviation of HV values of HypE and MODEhv. The numbers shown in boldface indicate better or comparable performance with respect to Wilcoxon sign test. Figure 4 gives the box plots of HV values of HypE and MODEhv in instances with five, seven, ten, and twenty-five objectives respectively.

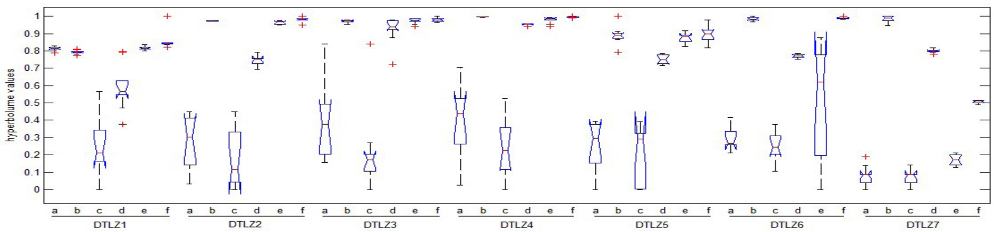
According to Table 2, MODEhv is slightly better than HypE in problems with five objectives, where MODEhv performs better in DTLZ1, DTLZ2, DTLZ4 and DTLZ6, HypE exceeds MODEhv in DTLZ5, and they are comparable in DTLZ3 and DTLZ7. In instances with seven, ten and twenty-five objectives, MODEhv obtains better performance in 17 out of 21 comparisons, and obtains comparable performance with respect to HypE in the other 4 out of 21 comparisons. Thus, it can be claimed that MODEhv performs better than HypE in problems with seven to twenty-five objectives. Furthermore, it is worth noting that MODEhv is more robust than HypE in most cases (19 out of 28 comparisons), as shown in Figure 4. The results above clearly show that the performance of MODEhv is better than that of HypE, which means that the modified DE paradigm and the additional techniques in MODEhv are rather effective for MaOPs.



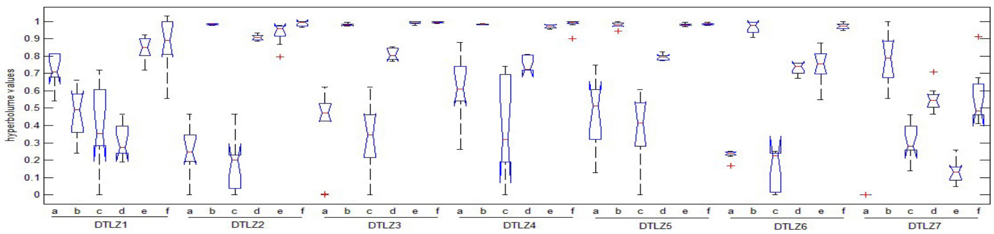
Five objectives



Seven objectives



Ten objectives



Twenty-five objectives

FIGURE 2. Box plots of HV values of (a) NSGA-II, (b) IBEA, (c) SMPSO, (d) IBMOLS, (e) HypE and (f) MODEhv in problems with five objectives, seven objectives, ten objectives and twenty-five objectives respectively.

It should be noted that Monte Carlo simulation is employed to estimate the HV value in MODEhv, and 10 000 samples are implemented in per simulation for saving computational time. According to [13], 10 000 samples per simulation will achieve an estimation accuracy of 96.9% and only 107 samples per simulation could achieve an estimation accuracy of 100%, so the performance of MODEhv is expected to be lower than its authentic performance.

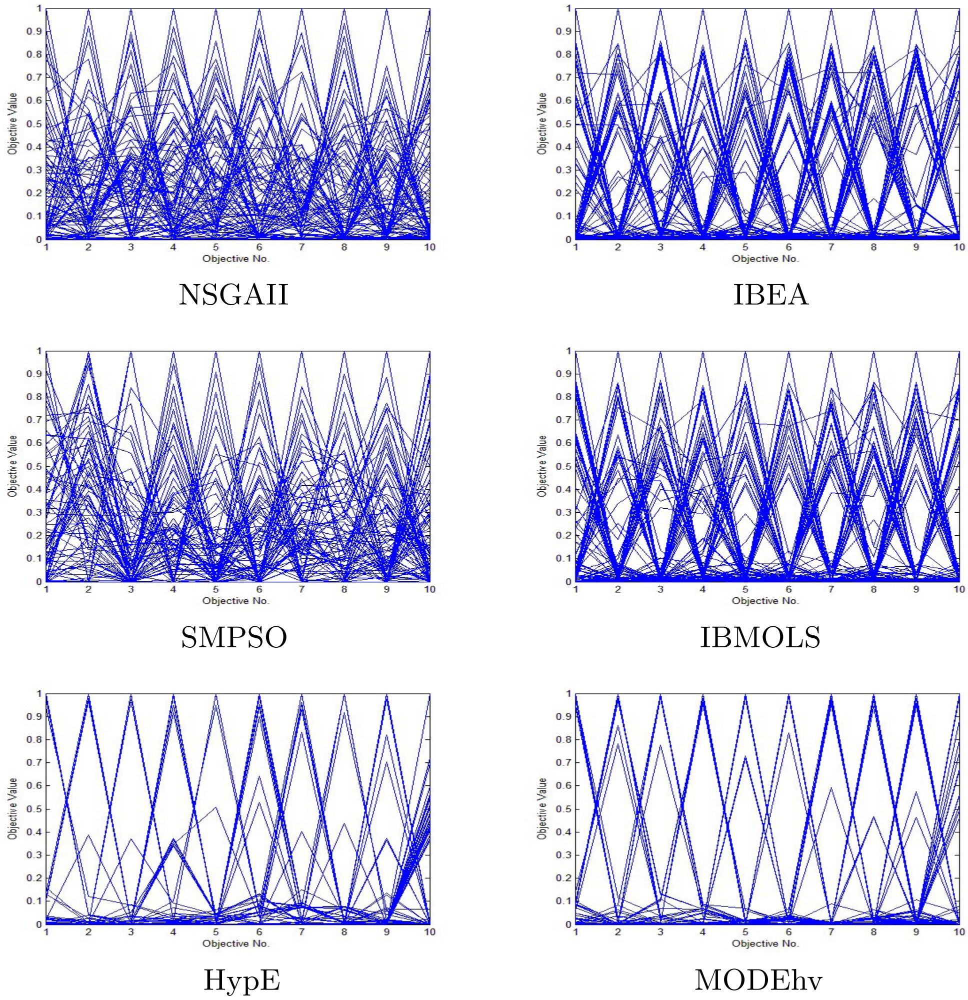


FIGURE 3. Parallel coordinates of non-dominated solutions generated by NSGA-II, IBEA, SMPSO, IBMOLS, HypE and MODEhv in DTLZ2 with ten objectives.

5.4. Discussion about parameter configuration

There are two tunable parameters α_1 (as $\alpha_1 = \alpha_2 = \alpha_3$, $\alpha_4 = 1 - 3 * \alpha_1$) and β in MODEhv. To justify the reasonability of the parameters given in MODEhv, the influences of different feasible parameter configurations on MODEhv are investigated.

To examine the effect of α_1 on MODEhv, three MODEhv versions with α_1=0.1, 0.15 and 0.2 are tested by DTLZ1-4 with five, seven and ten objectives, β is set to 0.75 constantly for fair comparison. Each instance is executed for 20 times, the average HV value of each MODEhv version in per instance is shown in Figure 5. It can be seen that MODEhv with $\alpha_1 = 0.15$ performs best in most cases. MODEhv with $\alpha_1 = 0.1$ is incomparable with $\alpha_1 = 0.15$ in DTLZ1 and DTLZ2, and ranks second in DTLZ3 and DTLZ4 except for ten-objective instances. These results indicate that the configuration of MODEhv ($\alpha_1 = 0.15$) is reasonable when solving different problems with different number of objectives.

To investigate the influence of parameter β on MODEhv, four MODEhv versions with $\beta = 0.7$, 0.75, 0.8 and 0.85 are tested by DTLZ1-4 with five, seven and ten objectives, here α_1 is set to 0.15 constantly for fair

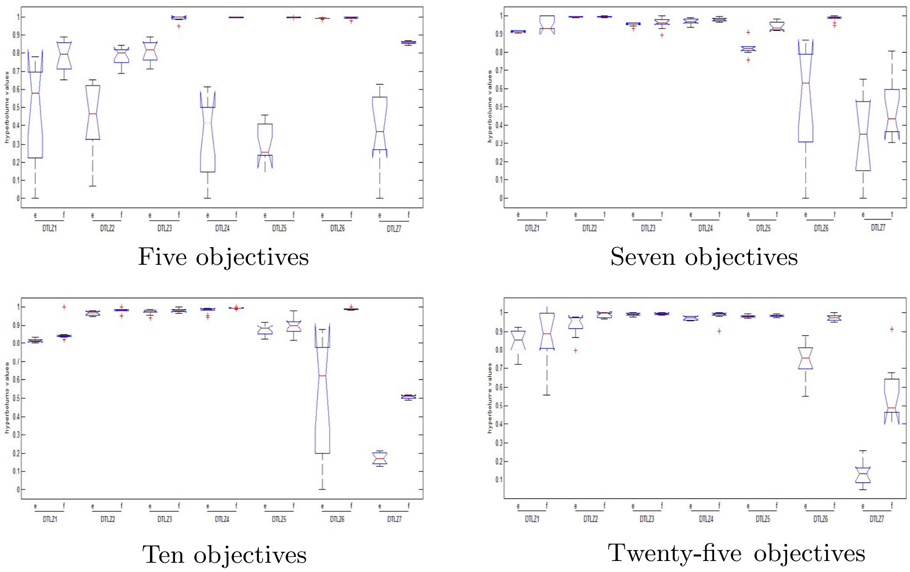


FIGURE 4. Box plots of HV values of (e) HypE and (f) MODEhv in problems with five objectives, seven objectives, ten objectives and twenty-five objectives respectively.

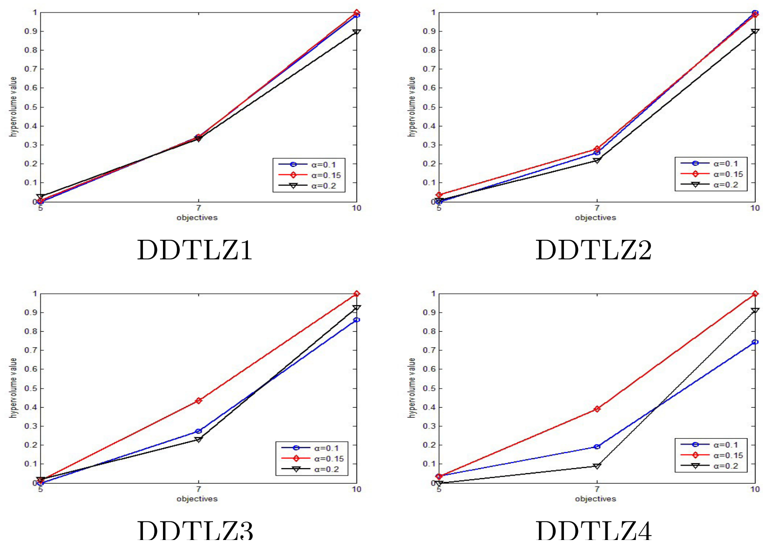


FIGURE 5. Influence of parameter α_1 on MODEhv in DTLZ1, DTLZ2, DTLZ3 and DTLZ4 with five-, seven- and ten objectives respectively.

comparison. Each instance is executed for 20 times, the average HV value of each MODEhv version in per instance is shown in Figure 6. It can be observed that MODEhv is not sensitive to β in problems with seven and ten objectives, mainly because solutions are prone to be nondominated with each other in MaOPs [19], almost all solutions will be assigned to Pareto front 1 following the Nondominated Sorting method [8], thus N solutions should be deleted from Pareto front 1 without taking account of β. In five-objective scenarios,

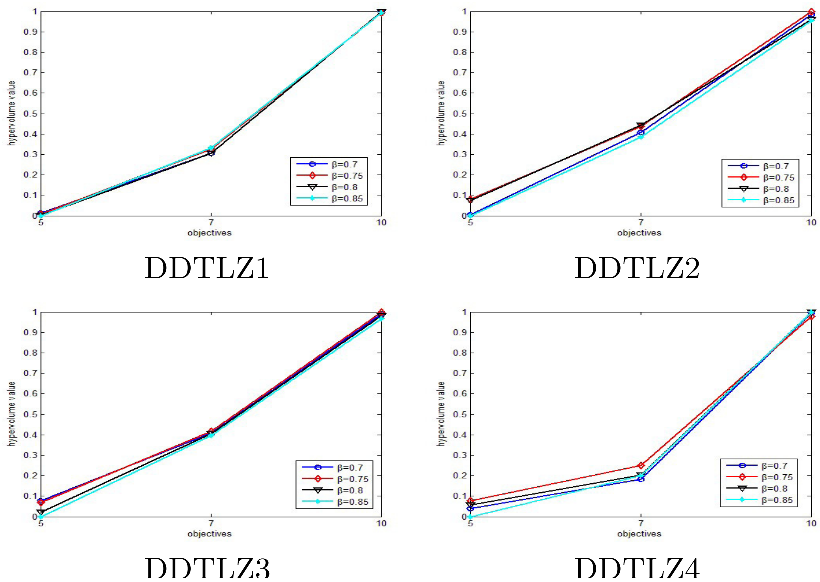


FIGURE 6. Influence of parameter β on MODEhv in DTLZ1, DTLZ2, DTLZ3 and DTLZ4 with five-, seven- and ten- objectives respectively.

$\beta = 0.75$ outperforms other configurations in most instances. Thus, it could be calmed that $\beta = 0.75$ is a reasonable configuration for MODEhv.

6. Conclusion

In this study, a new differential evolutionary algorithm based on hypervolume is designed for continuous MaOPs. In the proposed algorithm, a modified DE paradigm with automatic parameter configuration strategy is used, which better balances exploration and exploitation of the algorithm with respect to commonly used DE paradigms. Moreover, to avoid the invalidation of Pareto dominance in MaOPs, the proposed algorithm employs a hypervolume-based fitness assignment method in choosing parameter solutions in variation process, and uses a truncate strategy based on the minimum hypervolume loss in archive updating. Finally, to avoid the highest ranked solution filling up the entire population, a threshold-based technique is used to maintain the diversity of solutions in archive. According to the empirical analysis of the experimental result, the proposed algorithm obtains competitive results in problems with five objectives and outperforms the other tested algorithms in higher dimension problems. Thus, it can been seen that the proposed algorithm is effective in continuous MaOPs.

Future works include designing proper mutation and crossover operators to further accelerate convergence and maintain diversity, employing different selection operators to improve the diversity of solutions, and also extending the availability of the proposed algorithm to specific problems such as multimodal or combinational optimization problems.

References

[1] M. Ali, P. Siarry and M. Pant, An efficient differential evolution based algorithm for solving multi-objective optimization problems. *Europ. J. Oper. Res.* **217** (2012) 404–416.

[2] H. Abbass, R. Sarker and C. Newton, PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. Evolutionary Computation. *Proc. of the 2001 Congress* **2** (2001) 971–978.

[3] J. Bader and E. Zitzler, HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Comput.* **19** (2011) 45–76.

[4] S. Bandyopadhyay and A. Mukherjee, An Algorithm for Many-Objective Optimization with Reduced Objective Computations: A Study in Differential Evolution. *IEEE Trans. Evolut. Comput.* **19** (2015) 400–413.

[5] M. Basseur, A. Liefooghe and K. Le, The efficiency of indicator-based local search for multi-objective combinatorial optimisation problems. *J. Heuristics* **18** (2012) 263–296.

[6] D.W. Corne and J.D. Knowles, Techniques for highly multiobjective optimisation: some nondominated points are better than others. *Proc. of the 9th annual conference on Genetic and evolutionary computation*. ACM (2007) 773–780.

[7] O. Chikumbo, E. Goodman and K. Deb, Approximating a multi-dimensional pareto front for a land use management problem: A modified moea with an epigenetic silencing metaphor. *IEEE Congress on Evolutionary Computation*. IEEE (2012) 1–9.

[8] K. Deb, A. Pratap and S. Agarwal, A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation. *IEEE Trans.* **6** (2002) 182–197.

[9] S. Das and P.N. Suganthan, Differential evolution: a survey of the state-of-the-art. Evolutionary Computation. *IEEE Trans.* **15** (2011) 4–31.

[10] K. Deb, L. Thiele and M. Laumanns, Scalable multi-objective optimization test problems. *Proc. of the Congress on Evolutionary Computation (CEC-2002), Honolulu, USA* (2002) 825–830.

[11] K. Deb and D.K. Saxena, On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. Kangal report (2005) 2005011.

[12] G. Fu, Z. Kapelan and J.R. Kasprzyk, Optimal design of water distribution systems using many-objective visual analytics. *J. Water Resources Plan. Manag.* **139** (2012) 624–633.

[13] S. Jiang and Z. Cai, A new differential evolution for multiobjective optimization by uniform design and minimum reduce hypervolume. *Natural Computing*. Springer Jpn (2010) 199–208.

[14] X. He, C. Dai and Z. Chen, Many-Objective Optimization Using Adaptive Differential Evolution with a New Ranking Method. *Math. Probl. Eng.* (2014) 259–473.

[15] I. Giagkiozis, R.C. Purshouse and P.J. Fleming, An overview of population-based algorithms for multi-objective optimisation. *Inter. J. Syst. Sci.* **46** (2015) 1572–1599.

[16] H. Li and Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *Evolut. Comput., IEEE Trans.* **13** (2009) 284–302.

[17] K. Narukawa and T. Rodemann, Examining the performance of evolutionary many-objective optimization algorithms on a real-world application[C]. Genetic and Evolutionary Computing (ICGEC). *Sixth Inter. Confer. IEEE* (2012) 316–319.

[18] A.J. Nebro, J.J. Durillo and J. Garcia−Nieto, Smpso: A new pso-based metaheuristic for multi-objective optimization. Computational intelligence in miulti-criteria decision-making. MCDM'09. *IEEE Symposium* (2009) 66–73.

[19] R.C. Purshouse and P.J. Fleming, On the evolutionary optimization of many conflicting objectives. Evolutionary Computation. *IEEE Trans.* **11** (2007) 770–784.

[20] K.E. Parsopoulos, D.K. Tasoulis and N.G. Pavlidis, Vector evaluated differential evolution for multiobjective optimization. *IEEE Congress on Evolutionary Computation* (2004) 204–211.

[21] A. Ponsich, A.L. Jaimes and C. Coello, A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. Evolutionary Computation. *IEEE Trans.* **17** (2013) 321–344.

[22] R.G.D. Steel, J.H. Torrie and D.A. Dickey, Principles and procedures of statistics a biometrical approach. WCB/McGraw-Hill (1997).

[23] R. Storn and K.Price, Differential evolution¨Ca simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimiz.* **11** (1997) 341–359.

[24] H. Ishibuchi, N. Tsukamoto and Y. Nojima, Evolutionary many-objective optimization. Genetic and Evolving Systems. *GEFS. 3rd International Workshop on IEEE* (2008) 47–52.

[25] T. Wagner, N. Beume and B. Naujoks, Pareto-, aggregation-, and indicator-based methods in many-objective optimization. Evolutionary multi-criterion optimization. Springer Berlin/Heidelberg (2007) 742–756.

[26] S. Kukkonen and J. Lampinen, GDE3: The third evolution step of generalized differential evolution. *Evolutionary Computation. IEEE Congress on Evolutionary Computation* **1** (2005) 443–450.

[27] G. Wu, R. Mallipeddi and P.N. Suganthan, Differential evolution with multi-population based ensemble of mutation strategies. *Inform. Sci.* **329** (2016) 329–345.

[28] M. Črepinšek, S.H. Liu and M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)* **45** (2013) 35.

[29] E. Zitzler, L. Thiele and J. Bader, On set-based multiobjective optimization. *IEEE Trans. Evolut. Comput.* **14** (2010) 58–79.

[30] E. Zitzler and S. Knzli, Indicator-based selection in multiobjective search. Parallel Problem Solving from Nature-PPSN VIII. Springer Berlin Heidelberg (2004) 832–842

[31] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications. Ithaca: Shaker (1999).

[32] E. Zitzler, L. Thiele and M. Laumanns, Performance assessment of multiobjective optimizers: an analysis and review. *Evolut. Comput., IEEE Trans.* **7** (2003) 117–132.

[33] A. Zhou, B. YQu and H. Li, Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Comput.* **1** (2011) 32–49.

[34] E. Zitzler, M. Laumanns and L. Thiele, Improving the performance of the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich (2001).