

Research Article

Many-Objective Optimization Using Adaptive Differential Evolution with a New Ranking Method

Xiaoguang He,¹ Cai Dai,² and Zehua Chen³

¹ School of Basic Military Education, Engineering University of Chinese People's Armed Police Force, Xi'an 710068, China

² School of Computer Science and Technology, Xidian University, Xi'an 710071, China

³ Yingshang Third School, Fuyang 236200, China

Correspondence should be addressed to Cai Dai; daicai8403@hotmail.com

Received 6 June 2014; Revised 19 July 2014; Accepted 26 July 2014; Published 12 August 2014

Academic Editor: Erik Cuevas

Copyright © 2014 Xiaoguang He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pareto dominance is an important concept and is usually used in multiobjective evolutionary algorithms (MOEAs) to determine the nondominated solutions. However, for many-objective problems, using Pareto dominance to rank the solutions even in the early generation, most obtained solutions are often the nondominated solutions, which results in a little selection pressure of MOEAs toward the optimal solutions. In this paper, a new ranking method is proposed for many-objective optimization problems to verify a relatively smaller number of representative nondominated solutions with a uniform and wide distribution and improve the selection pressure of MOEAs. After that, a many-objective differential evolution with the new ranking method (MODER) for handling many-objective optimization problems is designed. At last, the experiments are conducted and the proposed algorithm is compared with several well-known algorithms. The experimental results show that the proposed algorithm can guide the search to converge to the true PF and maintain the diversity of solutions for many-objective problems.

1. Introduction

Multiobjective evolutionary algorithms (MOEAs) are a kind of effective methods for solving multiobjective problems [1, 2]. Almost all well-known and frequently used MOEAs, which have been proposed in the last twenty years [3–5], are based on Pareto dominance. Such Pareto dominance-based algorithms usually work well on problems with two or three objectives but its searching ability is often severely degraded by the increased number of objectives [6]. This is due to the fact that most solutions in a population could be nondominated solutions under many objectives even in the early stages of evolution for many-objective optimization problems. When almost all solutions in a population are nondominated solutions, Pareto dominance-based fitness evaluation almost cannot generate any selection pressure toward the Pareto front (PF). Therefore, how to increase the selection pressure toward the PF is critical for the many-objective optimization algorithm.

In the literature, there are mainly three categories which are used to MOEAs to deal with many-objective optimization

problems. The first category uses an indicator function, such as the hypervolume [7–9], as the fitness function. This kind of algorithms is also referred to as IBEAs (indicator-based evolutionary algorithms), and their high search ability has been demonstrated in the literature [10]. Bader and Zitzler [11] propose a fast hypervolume-based many-objective optimization algorithm which uses Monte Carlo simulation to quickly approximate the exact hypervolume values. However, one of their main drawbacks is the computation time for the hypervolume calculation which exponentially increases with the number of objectives.

The second category utilizes the scalarizing functions to deal with the many-objective problem. According to the literatures [12–14], scalarizing function-based algorithms could better deal with many-objective problems than the Pareto dominance-based algorithms. The main advantage of scalarizing function-based algorithms is the simplicity of their fitness evaluation which can be easily calculated even if the number of objectives is large. The representative MOEA in this category is MOEA/D [15] (multiobjective evolutionary

TABLE 1: Multiobjective benchmark functions.

	Objective function	Domain
DTLZ1	$f_1 = 0.5x_1x_2 \cdots x_{m-1}(1 + g(x))$	[0, 1] ⁿ
	$f_2 = 0.5x_1x_2 \cdots (1 - x_{m-1})(1 + g(x))$	
	\vdots	
	$f_{m-1} = 0.5x_1(1 - x_2)(1 + g(x))$	
	$f_m = 0.5(1 - x_1)(1 + g(x))$	
	$g(x) = 100(n - 2) + 100 \sum_{i=3}^n \{(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\}$	
DTLZ3	$f_1 = \cos(0.5\pi x_1) \cos(0.5\pi x_2) \cdots \cos(0.5\pi x_{m-1})(1 + g(x))$	[0, 1] ⁿ
	$f_2 = \cos(0.5\pi x_1) \cos(0.5\pi x_2) \cdots \sin(0.5\pi x_{m-1})(1 + g(x))$	
	\vdots	
	$f_{m-1} = \cos(0.5\pi x_1) \sin(0.5\pi x_2)(1 + g(x))$	
	$f_m = \sin(0.5\pi x_1)(1 + g(x))$	
	$g(x) = 100(n - 2) + 100 \sum_{i=3}^n \{(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\}$	

algorithm based on decomposition), which works well on a wide range of multiobjective problems with many objectives, discrete decision variables, and complicated Pareto sets [16–18]. In MOEA/D [15], the uniformity of the used weighted vectors determines the uniformity of the obtained nondominated optimal solutions; however, the used weighted vectors in MOEA/D are not very uniform and the size N of these weighted vectors should satisfy the restriction $N = C_{H+m-1}^m$. Thus N cannot be freely assigned and it will increase nonlinearly with the increase of m , where m is the number of objectives and H is an integer. This restricts the application of MOEA/D to many-objective optimization problems (i.e., m is large). Therefore, for many-objective problems, how to set weight vectors is a very difficult but critical task.

The third category makes use of solution ranking methods. Specifically, solution ranking methods are used to discriminate among solutions in order to enhance the selection pressure toward the PF, which make sure the solutions are able to converge to the PF. Bentley and Wakefield [19] proposed ranking composition methods which extract the separated fitness of every solution into a list of fitness values for each objective. Ikeda et al. [20] proposed a relaxed form of dominance (RFD) to deal with what they called dominance resistant solutions, that is, solutions that are extremely inferior to others in at least one objective but are hardly dominated. Farina and Amato [21] proposed a dominance relation which takes into account the number of objectives where a solution is better, equal, and worse than another solution. Sato et al. [22] proposed a method to strengthen or weaken the selection process by expanding or contracting the solutions' dominance area.

In this paper, a new ranking approach for many-objective problems is proposed to rank the obtained solutions. In this approach, for each solution, a small number of virtual objective vectors which are evenly distributed around the objective vector of the solution are generated to determine whether the solution is a nondominance solution. The selection pressure, the diversity of solutions, and time consumption are taken into consideration by the ranking method. Moreover, a multiobjective DE algorithm based on the new ranking

method is designed to solve many-objective optimization problems.

The remainder of this paper is organized as follows. Section 2 introduces the new ranking method. The proposed many-objective algorithm is proposed in Section 3. Section 4 shows the experimental results of the proposed algorithm. Finally, Section 5 gives the conclusions and future works.

2. The New Ranking Method

A multiobjective optimization problem can be formulated as follows [23]:

$$\begin{aligned}
 \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\
 \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, 2, \dots, q \\
 & h_j(x) = 0, \quad j = 1, 2, \dots, p,
 \end{aligned} \tag{1}$$

where $x = (x_1, \dots, x_n) \in X \subset R^n$ is called decision variable and X is n -dimensional decision space. $f_i(x)$ ($i = 1, \dots, m$) is the i th objective to be minimized, $g_i(x)$ ($i = 1, 2, \dots, q$) defines i th inequality constraint, and $h_j(x)$ ($j = 1, 2, \dots, p$) defines j th equality constraint. Furthermore, all the constraints determine the set of feasible solutions which are denoted by Ω . To be specific, we try to find a feasible solution $x \in \Omega$ minimizing each objective function $f_i(x)$ ($i = 1, \dots, m$) in F (Table 1).

Definition 1 (Pareto dominance). Pareto dominance between solutions $x, z \in \Omega$ is defined as follows. If

$$\begin{aligned}
 \forall i \in \{1, 2, \dots, m\} \quad & f_i(x) \leq f_i(z) \\
 \wedge \exists i \in \{1, 2, \dots, m\} \quad & f_i(x) < f_i(z)
 \end{aligned} \tag{2}$$

are satisfied, x dominates (Pareto dominate) z (denoted by $x > z$).

For many-objective problems, to determine whether a solution is a nondominated one by using Pareto dominance, all other solutions should be used to make the comparisons

of objectives one by one among all objectives, and a lot of comparisons (thus a large amount of computation) will be needed. Also, Pareto dominance will result in a lot of nondominated solutions. In this work, a new ranking method is designed to determine whether a solution is a nondominated solution by only using a small number of virtual objective vectors. For each solution x , m points in objective space are generated on the following surface:

$$f_1^P + f_2^P + \dots + f_m^P = R^P, \quad (3)$$

where $P > 0$ and it controls the shape and size of the surface; $R = \min\{\|F(x)\|_P \mid x \in \text{POP}\}$, where POP is the set of the current solutions. Before the m points are generated, m vectors ($D^i, i = 1, \dots, m$) are firstly generated. The i th vector $D^i = (d_1, \dots, d_m)$ ($i = 1, \dots, m$) is generated by optimizing the following formula:

$$\begin{aligned} \min \quad & \sum_{j \neq i} (d_j - H * o_j)^2 \\ \text{s.t} \quad & \sum_{j=1}^m d_j = H \\ & d_j \in \{0, 1, \dots, H\}, \quad j = 1, \dots, m \\ & \begin{cases} d_i = 0 & \text{if } o_i = 0 \\ H * o_i - 1 \leq d_i < H * o_i & \text{else,} \end{cases} \end{aligned} \quad (4)$$

where $o = (o_1, \dots, o_m) = F(x) / \sum_{i=1}^m f_i(x)$ and H is an integer. After D^i is obtained, the i th virtual vector is generated by the following formula:

$$V^i = R * \frac{D^i}{\|D^i\|_P}. \quad (5)$$

Note that these m virtual vectors have the same and the smallest norm (P -norm) among all obtained solutions and distribute evenly around the objective vector of the solution x . Moreover, for any virtual vector, for example, V^i ($i = 1, \dots, m$), its i th component is not larger than $f_i(x)$ and the i th component of D^i is not larger than $H o_i$, which indicate that the virtual vector V^i more likely dominates the solution $F(x)$ than other virtual vectors whose corresponding i th component of D^i is larger than $H o_i$. Thus, if $F(x)$ is not dominated by these m virtual vectors, the solution x can be regarded as a nondominated solution (of course, it may be a dominated one, but it is more possible to be a nondominated one); otherwise, the solution x can be regarded as a dominated solution.

The parameter H plays a very important role in this method which controls the convergence and diversity of solutions. The larger the value of H is, the better the convergence of solutions is. However, if the surface and the true PF are in great difference, a large value of H will lead to the poor diversity. Therefore, the value of H should be traded off between the convergence and diversity of solutions.

The proposed ranking method has the following advantages.

- (1) For each solution x of the set POP whose size is N , this method only needs to make comparisons among $m + 1$ points, instead of N points ($m \ll N$), which can reduce much time consumption.
- (2) The value of i th component of V^i is not more than $f_i(x)$, and the angle of V^i and $F(x)$ is very small, which make $F(x)$ easily dominated by V^i . Thus, the number of nondominated solutions generated by the method is not too large, and the selection pressure can be enhanced.
- (3) In this method, maximum m points need additional space to store, which only takes a small amount of storage space.
- (4) This method can balance the convergence and diversity by adjusting the parameter H .

3. Algorithm

In this section, a many-objective optimization evolutionary algorithm is presented. The algorithm uses DE [24, 25] operator to generate offspring and the $(\mu + \lambda)$ selection scheme to generate the next population. DE is a good optimizer for continuous optimization. The trial vector generation comprises two operators: mutation and crossover operators. The mutation operator which is used in our algorithm is performed as follows:

$$u_i^t = x_{r_1}^t + L(x_{r_2}^t - x_{r_3}^t), \quad (6)$$

where t is the generation number; $x_{r_1}^t$ and $x_{r_2}^t$ are two random nondominated solutions among the current population POP(t); $x_{r_3}^t$ is a random solution among POP(t) except for $x_{r_1}^t$ and $x_{r_2}^t$; u_i^t is i th mutation offspring; L is the scale factor which is usually in $(0, 1]$.

The crossover operator is described as follows:

$$v_{i,j}^t = \begin{cases} u_{i,j}^t, & \text{if } \text{rand} \leq \text{CR} \text{ and } l_j \leq u_{i,j}^t \leq u_j \\ x_{i,j}^t, & \text{else,} \end{cases} \quad (7)$$

where CR is the crossover rate which is usually in $(0, 1)$; l_j and u_j are the lower bound and upper bound of the variable x_j , respectively; $v_i^t = (v_{i,1}^t, \dots, v_{i,n}^t)$ is the offspring.

In classical DE, the control parameters F and CR highly affect its performance. Different settings of those parameters influence the quality of offspring which are generated by the mutation and crossover. In this paper, a self-adjusted parameter control strategy is presented to balance exploration and exploitation. For each solution x^t , its two control parameters which are used to generate the offspring are determined as follows:

$$\begin{aligned} \text{CR} &= \begin{cases} 0.5, & \text{if } \text{rand} < 0.5 \\ e^{-2t/g_{\max}}, & \text{else,} \end{cases} \\ L &= \begin{cases} 0.5, & \text{if } \text{rand} < 0.5 \\ e^{-2t/g_{\max}}, & \text{else,} \end{cases} \end{aligned} \quad (8)$$

where g_{\max} is the maximal generation.

- (1) The population POP
- (2) Each solution of POP is firstly considered as a non-dominate solution
- (3) Determine the values of R and P
- (4) for each solution $x \in \text{POP}$ do
- (5) $i = 1$, while $i \leq m$
- (6) Obtained the i th vector D^i by optimizing the formula (4);
- (7) then set $V^i = R * D^i / \|D^i\|_p$
- (8) If V^i dominates $F(x)$ then
- (9) x is a dominate solution and set $i = m + 1$
- (10) else
- (11) $i = i + 1$
- (12) end if
- (13) end while
- (14) end for

ALGORITHM 1: The ranking algorithm.

Based on the above methods, a new multiobjective differential evolution based on the new ranking method is proposed (MODER) for many-objective optimization problems and the steps of the algorithm MODER are described as follows.

Step 1 (initialization). Randomly generate an initial population $\text{POP}(t)$ whose size is N and $t = 0$.

Step 2 (fitness). Solutions of $\text{POP}(t)$ are firstly divided into two sets (the set of nondominated solutions and the set of dominated solutions) by the proposed ranking method (Algorithm 1). For each set, the fitness value of each solution in the set is calculated by the crowding distance. Then N better solutions are selected from the population $\text{POP}(t)$ and put these solutions into the population POP. In this work, binary tournament selection is used.

Step 3 (new solutions). Apply (6)–(8) or simulated binary crossover [26] to the parent population POP to generate offspring. The set of all these offspring is denoted by O and its size is N .

Step 4 (generate $\text{POP}(t + 1)$). Solutions of $\text{POP}(t) \cup O$ are firstly divided into two sets by the proposed ranking method. For each set, the fitness value of each solution in the set is calculated by the crowding distance. Then N better solutions are selected from the population $\text{POP}(t) \cup O$ and put these solutions into the population $\text{POP}(t + 1)$. Let $t = t + 1$.

Step 5 (termination). If stop condition is satisfied, stop; otherwise, go to Step 2.

4. Experimental Studies

In order to validate our algorithm, MODER is fully compared with MOEA/D [17], NSGAI [3], and NSGAI based on contracting or expanding the solutions' dominance area [22] (denoted by NSGAI-CE) on DTLZ1 and DTLZ3 of the DTLZ family [27], each with 5–50 objectives.

4.1. Experimental Setting. The experiments are carried out on a personal computer (Intel Xeon CPU 2.53 GHz, 3.98 G RAM). The solutions are all coded as real vectors. Polynomial mutation [28] operators and differential evolution (DE) [24] are applied directly to real vectors in three algorithms, that is, MOEDR, MOEA/D, and NSGAI-CDAS. The distribution index and crossover probability in polynomial mutation [28] operators are set to 20 and $1/n$, respectively. The crossover rate and scaling factor in DE operator are set to 1.0 and 0.5, respectively. The aggregate function of MOEA/D is the Tchebycheff approach [15], and weight vectors are generated by using the uniform design method for MOEA/D and UREA/D. The number of the weight vectors in the neighborhood in MOEA/D is set to 20 for all test problems. The parameter of the ranking method H is set to 2, and in each generation, the value of P is self-adaptive determined by the following expression:

$$P = \begin{cases} k, & \text{if } R(1) < \frac{R(k)(1 + \sqrt{m})}{2} \\ 1, & \text{else,} \end{cases} \quad (9)$$

where $R(k) = \min\{\|F(x)\|_k \mid x \in \text{POP}\}$ and m is the number of objectives. In this algorithm, we set $k = 2$. The parameter of CE is set to 0.25 for NSGAI-CE. 20 independent runs are performed with population size of 100 for all these instances. The maximal number of function evaluations is set to 100000 for all test problems. The values of default parameters are the same as in the corresponding papers.

4.2. Experimental Measures. In order to compare the performance of the four compared algorithms quantitatively, the following three widely used performance metrics are adopted: generational distance (GD), inverted generational distance (IGD) [29], and Wilcoxon rank-sum test [30], where GD measures how far the obtained Pareto front is away from the true Pareto front, which allows us to observe whether the algorithm can converge to the true PF. If GD is equal to 0, all points of the obtained PF belong to the true PF. IGD measures how far the true PF is away from the obtained PF, which shows whether points of the obtained PF are evenly

TABLE 2: IGD, GD, and HV obtained by MODER, MOEA/D, NSGAI, and NSGAI-CE on DTLZ1.

		IGD		GD	
		Mean	Std.	Mean	Std.
DTLZ1-5	MODER	0.0719	0.0067	0.0311	0.0025
	MOEA/D	0.0756(+)	0.0074	0.0317(+)	0.0035
	NSGAI	23.789(+)	15.261	233.69(+)	20.501
	NSGAI-CE	0.2989(+)	0.1023	0.0124(-)	0.0202
DTLZ1-10	MODER	0.0921	0.0146	0.1160	0.0075
	MOEA/D	0.1045(+)	0.0081	0.1141(=)	0.0242
	NSGAI	25.065(+)	6.9643	343.45(+)	10.444
	NSGAI-CE	0.3246(+)	0.0969	0.0055(-)	0.0079
DTLZ1-15	MODER	0.0968	0.0063	0.1925	0.0103
	MOEA/D	0.1141(+)	0.0101	0.1340(-)	0.0198
	NSGAI	15.866(+)	11.499	222.14(+)	5.2899
	NSGAI-CE	0.4152(+)	0.0942	0.5510(+)	0.0457
DTLZ1-20	MODER	0.1523	0.0613	0.2289	0.0056
	MOEA/D	0.1162(-)	0.0176	0.1542(-)	0.0133
	NSGAI	32.596(-)	13.770	373.34(+)	5.1034
	NSGAI-CE	0.4197(-)	0.0604	0.1218(-)	0.1244
DTLZ1-25	MODER	0.2030	0.0585	0.2679	0.0302
	MOEA/D	0.1599(-)	0.0881	0.2165(-)	0.1604
	NSGAI	37.273(+)	21.605	389.85(+)	6.1814
	NSGAI-CE	0.4694(+)	0.0558	0.1237(-)	0.1144
DTLZ1-50	MODER	0.3180	0.0392	0.2865	0.0845
	MOEA/D	0.1623(-)	0.0142	0.1570(-)	0.0290
	NSGAI	33.276(+)	13.482	364.25(+)	9.7122
	NSGAI-CE	1.1990(+)	2.3179	1.5583(+)	3.5537

“+” means that MODER outperforms its competitor algorithm, “-” means that MODER is outperformed by its competitor algorithm, and “=” means that the competitor algorithm has the same performance as MODER.

distributed throughout the true PF. If IGD is equal to 0, the obtained PF contains every point of the true PF. In particular, GD and IGD indicators are used simultaneously to observe whether the solutions are distributed over the entire PF or the solutions concentrate in some regions of the true PF. In our experiments, 100000 points which are generated by using the uniform design [31] for all these test instances which are uniformly sampled on the true PF are used to calculate the metrics GD and IGD of solutions obtained by an algorithm.

Wilcoxon rank-sum test [30] is used in the sense of statistics to compare the mean IGD and GD of the compared algorithms. It tests whether the performance of MODER on each test problem is better than (“+”), same as (“=”), or worse than (“-”) that of the compared algorithms at a significance level of 0.05 by a two-tailed test. These results are given in Tables 2 and 3.

4.3. Comparisons of MODER with MOEA/D, NSGAI, and NSGAI-CE. In this section, some simulation results and comparisons that demonstrate the potential of MODER are presented, and the comparisons mainly focus on two aspects: (1) the ability of the ranking method to improve the selection pressure and (2) the ability of the ranking method to maintain the diversity of solutions.

Tables 2 and 3 show the mean and standard deviation of the GD metric and IGD metric obtained by these four algorithms for test problems with 5–50 objectives, respectively. DTLZ1-5 represents that the number of objectives of DTLZ1 is 5. It can be seen from Tables 2 and 3 that, for the IGD metric, MODER outperforms NSGAI and NSGAI-CE on all twelve test problems and outperforms MOEA/D on nine test problems. For the GD metric, MODER outperforms NSGAI on all twelve test problems, outperforms NSGAI-CE on eight test problems, outperforms MOEA/D on seven test problems, and performs worse than NSGAI-CE and MOEA/D on four test problems. These results indicate that MODER outperforms all three compared algorithms.

These results of Tables 2 and 3 also show that the mean values of GD and IGD obtained by NSGAI are the largest among the four algorithms, which indicate that NSGAI has the worst performance in the convergence to the true PF among these four algorithms and Pareto dominance has little selection pressure toward the true PF. For problem DTLZ1 with 20–50 objectives, the mean values of IGD obtained by MODER are slightly larger than those obtained by MOEA/D, which indicates that MOEA/D can better maintain the diversity of solutions than MODER for these problems. However, for problem DTLZ1 with 5–15 objectives, the mean values of IGD obtained by MODER are smaller than those obtained

TABLE 3: IGD, GD, and HV obtained by MODER, MOEA/D, NSGAI, and NSGAI-CE on DTLZ3.

		IGD		GD	
		Mean	Std.	Mean	Std.
DTLZ3-5	MODER	0.231	0.0168	0.0837	0.0040
	MOEA/D	0.2499(+)	0.0667	0.0869(+)	0.0071
	NSGAI	0.6093(+)	0.1186	0.8738(+)	0.1039
	NSGAI-CE	1.0903(+)	0.2406	0.0913(+)	0.0914
DTLZ3-10	MODER	0.4099	0.0171	0.3130	0.0126
	MOEA/D	0.4264(+)	0.0585	0.3219(+)	0.0182
	NSGAI	65.508(+)	14.659	812.75(+)	23.198
	NSGAI-CE	1.0413(+)	0.3021	152.583(+)	234.71
DTLZ3-15	MODER	0.4723	0.0106	0.5662	0.0436
	MOEA/D	0.6953(+)	0.0585	0.7038(+)	0.0703
	NSGAI	85.563(+)	18.166	937.31(+)	22.914
	NSGAI-CE	1.1584(+)	0.3419	20.217(+)	53.426
DTLZ3-20	MODER	0.5642	0.1296	0.8176	0.0746
	MOEA/D	0.7542(+)	0.0233	1.0025(+)	0.0512
	NSGAI	36.307(+)	17.207	573.32(+)	11.818
	NSGAI-CE	1.2466(+)	0.3100	0.9260(+)	0.5774
DTLZ3-25	MODER	0.5292	0.0340	0.9388	0.0508
	MOEA/D	0.7894(+)	0.1075	1.1898(+)	0.0433
	NSGAI	90.042(+)	111.47	931.18(+)	14.475
	NSGAI-CE	1.4141(+)	0.0002	6.7310(+)	15.187
DTLZ3-50	MODER	0.7507	0.0337	0.5076	0.0048
	MOEA/D	0.9589(+)	0.2665	1.1179(+)	0.4531
	NSGAI	71.523(+)	31.877	703.20(+)	284.21
	NSGAI-CE	0.9211(+)	0.4885	1.9211(+)	0.4885

“+” means that MODER outperforms its competitor algorithm, “-” means that MODER is outperformed by its competitor algorithm, and “=” means that the competitor algorithm has the same performance as MODER.

by MOEA/D, and for problem DTLZ1 with 5–50 objectives, the mean values of IGD obtained by MODER are smaller than those obtained by NSGAI-CE, which indicate that MODER has a better ability to maintain the diversity than NSGAI-CE and well maintains the diversity of obtained solutions. These also show that the new ranking method can better maintain the diversity of solutions than CE. For the convergence metric GD, the mean values of GD obtained by MODER are smaller than 0.29 for DTLZ1 with 5–50 objectives and are slightly larger than those obtained by MOEA/D. MOEA/D decomposes a multiobjective problem (MOP) into a number of scalar optimization subproblems and solves them simultaneously. The objective of each of these subproblems is an aggregation of all the objectives in the MOP under consideration. Single objective is more likely to converge to the optimal solution, and its degree of convergence is relatively high. These show that the convergence of solutions obtained by MODER to the true PF is good and the new ranking method can enhance the selection pressure toward the true PF. For problem DTLZ3 with 5–50 objectives, the mean values of IGD obtained by MODER are smaller than those obtained by MOEA/D and NSGAI-CE, which show that the diversity of solutions obtained by MODER is better than those obtained by MOEA/D and NSGAI-CE; the mean values of GD obtained by MODER are smaller than those

obtained by MOEA/D and NSGAI-CE, which indicate that the convergence of solutions obtained by MODER is better than those obtained by MOEA/D and NSGAI-CE. These imply that MODER can converge to the true PF and maintain the diversity of obtained solutions; even for higher dimensional objectives, its convergence and diversity performance are still good. In other words, the new ranking method not only can enhance the selection pressure toward the true PF but also can well maintain the diversity of solutions.

5. Conclusions

In this work, a new ranking method for many-objective optimization problems is proposed to decrease the number of nondominated solutions and reduce the number of comparisons in the process of verifying the nondominated solutions. For each solution x , the approach generates a small number of virtual vectors which are evenly distributed around the objective vector of the solution x and can determine whether x is a nondominated solution by only making the comparison between $F(x)$ and every virtual vector. Thus, this method not only uses a small quantity of comparisons when determining the nondominated solutions, but also is able to contract the nondominated area of the solution to increase the selection

pressure toward the true PF. Moreover, a multiobjective adaptive differential evolution algorithm based on the new ranking method is designed for many-objective problems. The algorithm was tested on problems with up to 50 objectives. Compared with the existing algorithms, simulation results show that the proposed algorithm is able to maintain the diversity of the obtained solutions and has good convergence ability on problems with large number of many objectives. Future works include expanding the experiments to other problems and search another crossover operator which is more suitable for many-objective problems.

Conflict of Interests

The authors have declared that no conflict of interests exists.

References

- [1] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer, New York, NY, USA, 2006.
- [2] I. Kokshenev and A. Padua Braga, "An efficient multi-objective learning algorithm for RBF neural network," *Neurocomputing*, vol. 73, no. 16–18, pp. 2799–2808, 2010.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [4] A. Nebro, J. Durillo, J. Garcia-Nieto, C. A. C. Coello, F. Luna, and E. Alba, "SMPSO: a new PSO-based metaheuristic for multi-objective optimization," in *Proceedings of the IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM '09)*, pp. 66–73, 2009.
- [5] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [6] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.
- [7] T. Friedrich, C. Horoba, and F. Neumann, "Multiplicative approximations and the hypervolume indicator," in *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference (GECCO '09)*, pp. 571–578, July 2009.
- [8] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: Optimal μ -distributions and the choice of the reference point," in *Proceedings of the 10th ACM SIGEVO Conference on Foundations of Genetic Algorithms*, pp. 87–102, 2009.
- [9] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature—PPSN VIII*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 832–842, Springer, Berlin, Germany, 2004.
- [10] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in manyobjective optimization," in *Proceedings of the Evolutionary Multi-Criterion Optimization (EMO '07)*, vol. 4403 of *Lecture Notes in Computer Science*, pp. 742–756, Springer, 2007.
- [11] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [12] E. J. Hughes, "Evolutionary many-objective optimisation: many once or one many?" in *Proceeding of the IEEE Congress on Evolutionary Computation (CEC '05)*, vol. 1, pp. 222–227, Edinburgh, Scotland, September 2005.
- [13] E. J. Hughes, "MSOPS-II: a general-purpose many-objective optimiser," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 3944–3951, September 2007.
- [14] H. Ishibuchi and Y. Nojima, "Optimization of scalarizing functions through evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*, vol. 4403 of *Lecture Notes in Computer Science*, pp. 51–65, Springer, Berlin, Germany, 2007.
- [15] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [16] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 1820–1825, October 2009.
- [17] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [18] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D + uniform design: a new version of MOEA/D For optimization problems with many objectives," *Computers & Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.
- [19] P. J. Bentley and J. P. Wakefield, "Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms," in *Proceedings of the World Conference on Soft Computing in Design and Manufacturing*, pp. 231–240, 1997.
- [20] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?" in *Proceeding of the Congress on Evolutionary Computation*, vol. 2, pp. 957–962, Seoul, Korea, May 2001.
- [21] M. Farina and P. Amato, "A fuzzy definition of "optimality" for many-criteria optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics A—Systems and Humans*, vol. 34, no. 3, pp. 315–326, 2004.
- [22] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of mOEA," in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4403 of *Lecture Notes in Computer Science*, pp. 5–20, Springer, Heidelberg, Germany, 2007.
- [23] D. A. van Veldhuizen, *Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations [Ph.D. thesis]*, Department of Electrical & Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA, 1999.
- [24] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [25] M. Ali, M. Pant, and A. Abraham, "A modified differential evolution algorithm and its application to engineering problems," in *Proceedings of the International Conference on Soft Computing*

- and Pattern Recognition (SoCPaR '09)*, pp. 196–201, December 2009.
- [26] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, New York, NY, USA, Wiley, 2001.
- [27] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” Tech. Rep. 112, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
- [28] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.
- [29] C. A. C. Coello and N. C. Cortés, “Solving multiobjective optimization problems using an artificial immune system,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [30] S. Robert, J. Torrie, and D. Dickey, *Principles and Procedures of Statistics: A Biometrical Approach*, McGraw-Hill, New York, NY, USA, 1997.
- [31] K. Fang and Y. Wang, *Number-Theoretic Method in Statistics*, Chapman and Hall, London, UK, 1994.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

