Hindawi
Complexity
Volume 2020, Article ID 8353154, 7 pages
https://doi.org/10.1155/2020/8353154

Research Article

A Simplified Hypervolume-Based Evolutionary Algorithm for Many-Objective Optimization

Hong Ji and Cai Dai

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

Correspondence should be addressed to Cai Dai; cdai0320@snnu.edu.cn

Received 5 May 2020; Accepted 8 July 2020; Published 6 August 2020

Academic Editor: Toshikazu Kuniya

Evolutionary algorithms based on hypervolume have demonstrated good performance for solving many-objective optimization problems. However, hypervolume needs prohibitively expensive computational effort. This paper proposes a simplified hypervolume calculation method which can be used to roughly evaluate the convergence and diversity of solutions. The main idea is to use the nearest neighbors of a particular solution to calculate the volume as the solution's hypervolume value. Moreover, this paper improves the selection operator and the update strategy of external population according to the simplified hypervolume. Then, the proposed algorithm (SHEA) is compared with some state-of-the-art algorithms on fifteen test functions of CEC2018 MaOP competition, and the experimental results prove the feasibility of the proposed algorithm.

1. Introduction

Multiobjective optimization problems (MOPs) have been applied in numerous real-world applications. A minimized MOP which often has two or three objectives can be defined as follows [1]:

$$\begin{cases} \min F(x) = \left(f_1(x), f_2(x), \ldots, f_m(x)\right), \\ \text{s.t.} \quad x \in \Omega, \end{cases} \tag{1}$$

where $\Omega \subseteq R^n$ is an n-dimensional decision space; $x = (x_1, \ldots, x_n) \in \Omega$ is an n-dimensional decision variable; $F: \Omega \longrightarrow R^m$ $(m = 2 \text{ or } 3)$ contains m interconflicting objective functions.

In the last few decades, multiobjective evolutionary algorithms (MOEAs) [2–5] are proposed to solve MOPs. However, when solving MOPs with more than three objectives which can also be recognized as many-objective optimization problems (MaOPs) [6], these MOEAs encounter challenges. First of all, the proportion of nondominated solutions in candidate solutions rises steeply with the increasing number of objectives, which severely deteriorates the selection pressure toward PF. Secondly, the size of population cannot be arbitrarily large in consideration of computational efficiency. But limited number of solutions is probably far away from each other in high-dimensional objective space, causing the offspring to stay away from parents. Lastly, the computational complexity of performance metrics grows exponentially with the increasing number of objectives.

To solve these problems, there are three main categories of many-objective evolutionary algorithms (MaOEAs). The first category is based on the modified dominance relationship [7–9] to enhance the selection pressure to PF. This kind of idea has been widely employed and proved considerable improvement. However, these approaches need more efforts in designing diversity maintenance mechanism to ensure diversity.

The second category uses decomposition-based method to solve MaOPs. The main idea is to decompose a many-objective optimization problem into a set of subproblems and optimize them collaboratively. The most representative algorithms are MOEA/*D* [10] and its variants [11–13]. And there are also some other methods based on decomposition such as MOEA/D-M2M [14] and DBEA [15] [16–18]. These approaches are adept in diversity maintenance and avoiding local optimum but ineffective to address highly irregular PFs.

The third approach is indicator-based evolutionary algorithms. Indicators such as hypervolume [19] weigh both convergence and diversity of solutions to enhance the selection pressure and guide the search to PF. IBEA [20], SMS-EMOA [21], and HypE [22] are three classical indicator-based evolutionary algorithms. Unfortunately, the computational cost becomes excessively expensive because of the high computational complexity.

So the key question is how to reduce the computational complexity and keep the advantages of the hypervolume indicator at the same time. The major contributions of this paper can be summarized as follows.

(1) A simplified hypervolume calculation method is proposed to roughly evaluate the convergence and diversity of solutions

(2) To enhance the quality of offspring, a new selection operator based on the simplified hypervolume is proposed to choose excellent parents

(3) The simplified hypervolume together with non-domination is used in the new update strategy of external population to store solutions with good convergence and diversity

In the remainder of this paper, Section 2 describes the proposed algorithm. Then, Section 3 mainly presents experimental results and related analysis of the proposed algorithm. At last, conclusions are given in Section 4.

2. The Proposed Algorithm

A simplified hypervolume-based evolutionary algorithm for many-objective optimization (SHEA) is proposed to solve MaOPs. The core part of this paper is a new hypervolume calculation method to roughly evaluate the convergence and diversity of solutions. Furthermore, this new hypervolume is used to improve selection operator and update strategy.

2.1. Simplified Hypervolume. To get the hypervolume value, the normalized population P is sorted by each objective function value. For each solution of each objective function value sequence, the reference point is the maximum of the two points on either side of the particular solution.

Thereafter, the volume between the particular solution and the reference point is calculated. As for the boundary solutions, just calculate the volumes between these boundary solutions and its adjacent solutions and remove the terms when the objective function value of x_i is boundary value. So the calculation formula of v_i^j is (2). Figure 1 shows the calculation of v_i^j, and to make it easier to understand, the MOP in Figure 1 has only two objectives.

Then, the minimum of the particular solution's volumes for all objectives is kept as the hypervolume value (3):

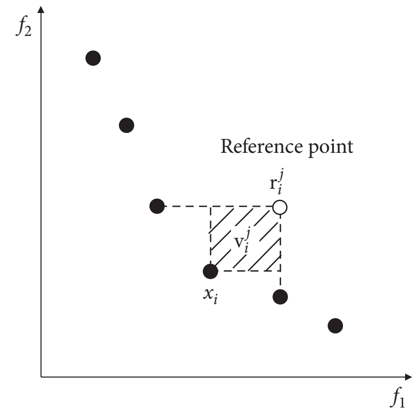

Figure 1: v_i^j calculation: the shadow is volume between x_i and the reference point r_i^j on objective j.

$$v_i^j = \sqrt[L]{\prod_{k\in L}\left\{f_k\left(r_i^j\right) - f_k\left(x_i\right) + \partial\right\}}, \tag{2}$$

$$\operatorname{shv}\left(x_i\right) = \min_{1\le j\le m} v_i^j, \tag{3}$$

where ∂ is a small positive number, $L = \{k \mid f_k(x_i) \neq \max\{f_k(x) | x \in P\}\}$, r_i^j $(i = 1, \ldots, N;\ j = 1, \ldots, m)$ and v_i^j $(i = 1, \ldots, N; j = 1, \ldots, m)$ are reference point and volume of the ith solution for the jth objective.

When x_i is sparse which means the adjacent solutions are far from x_i, v_i^j is large. As for convergence, when each function of x_i is small which means x_i is far from r_i^j, v_i^j is large. So, the convergence and diversity of x_i are better when $\operatorname{shv}(x_i)$ is larger.

2.2. Selection Operator. The new selection operator aims to choose parents with good convergence and diversity to generate high quality offspring in differential evolution [23]:

$$\mathrm{DE}\begin{cases} x_i + F\left(x_i^{r1} - x_i^{r2}\right), & \text{rand} \le \mathrm{CR}, \\ x_i, & \text{rand} > \mathrm{CR}, \end{cases} \tag{4}$$

where x_i^{r1} $(i = 1, \cdots, n)$ means the ith dimension of x^{r1} in decision space; CR is crossover rate; and x_i^{r1} and x_i^{r2} are selected in T nearest neighbors of solution x.

The new selection operator calculates the non-dominated neighbors' hypervolume and retains the top H solutions. To accelerate convergence, x_i^{r1} selects the minimum solution of Tchebycheff function in H retained solutions:

$$\underset{x\in\Omega}{\text{minimize}}\ g^{\text{te}}(x \mid \lambda, z) = \max_{1\le i\le m}\left\{\lambda_i\left|f_i(x) - z_i\right|\right\}, \tag{5}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is a given weight vector and $z = (z_1, \ldots, z_m)$ is a reference point with $z_i = \min\{f_i(x) \mid x \in \Omega\}$.

Then, randomly choose x_i^{r2} in neighbors except for x_i^{r1}.

Input:
MaOP(1)
A stopping criterion
N: the number of weight vectors
EN: the number of external population
T: the number of weight vectors in the neighborhood of each weight vector, $0 < T < N$
H: the number of the solutions with largest hypervolume selected in neighbors, $0 < H < T$
$\lambda^1, \lambda^2, \cdots, \lambda^N$: a set of N uniformly distributed weight vectors
Output: External population EP
Initialization: Generate an initial population $O = \{x^1, x^2, \ldots, x^N\}$ randomly; set EP = O; determine $Z = (z_1, \ldots, z_m)$ by a problem-specific method; determine T closest weight vectors to each vector $B(i) = \{i_1, \ldots, i_T\}$, $(i = 1, \ldots, N)$
While the stopping criterion is not met **do**
  Calculate the proposed hypervolume of nondominated solutions.
  For $i = 1, \ldots, N$ **do**
    if rand $< J$ **then**
      $E = B(i)$
    else
      $E = O$
    end if
    Choose x_i^{r1} and x_i^{r2} from E according to the selection operator in Section 2.2.
    Use x_i^{r1} and x_i^{r2} to generate offspring x^{new}, and set $O = O \cup x^{\text{new}}$.
    Use x^{new} **to Update Z**: For $j = 1, \ldots, m$, if $z_j < \min\{f_j(x^{\text{new}}) \mid x^{\text{new}} \in O\}$, then set $z_j = \min\{f_j(x^{\text{new}}) \mid x^{\text{new}} \in O\}$.
  End for
  Set EP = EP $\cup$ O.
  Use the updated strategy to update O.
  Use the updated strategy of external population of Section 2.3 to update EP.
End while

Algorithm 1: The framework of the algorithm SHEA.

2.3. Update Strategy of External Population. The external population is adopted to store solutions with good convergence and diversity. The solutions with smaller hypervolume values of the nondominated solutions of population are deleted to maintain the number of the external population. And to retain diversity, m solutions in external population are selected in boundary solutions; others come from intermediate solutions.

2.4. Steps of the Proposed Algorithm. SHEA works as follows (Algorithm 1):

To update the population, the solutions in set O are sorted by nondomination and kept in set SP from the first nondomination rank to the last until the total number of SP is bigger than N. Then, the cosine of solutions in SP and weight vectors are calculated to classify each solution into the corresponding weight vector according to the maximum cosine. For each weight vector λ^i, when solutions in this kind are not empty, save the solution with minimum Tchebycheff function of modified version:

$$\underset{x\in\Omega}{\text{minimize}}\, g^{\text{te}}\left(x \mid \lambda^i, z\right) = \max_{1\le j\le m}\left\{\frac{\left|f_j(x) - z_j\right|}{\lambda_j^i}\right\}. \quad (6)$$

Otherwise, find the solution with the minimum Tchebycheff function in SP.

In the proposed algorithm, $2N + \text{EN}$ solutions (population, offspring, and external population), $B(i)$ $(i = 1, \ldots, N)$, and$N$ weight vectors need to be stored, so the space complexity is $O((2N + \text{EN}) * tn) + O(N * T) + O(N * m) = O(N^2)$ (in this paper, n, $T, m < N < \text{EN} \le 2N$). Therefore, the space complexity of SHEA is $O(N^2)$. The major computation of the proposed algorithm contains the selection operator and the update strategy of external population. Both of them use the simplified hypervolume, which needs $O(N * n)$ basic operations (i.e., +, −, ×, ÷, and comparison). So, the selection operator needs $O(N * n)$ basic operations to calculate the simplified hypervolume and $O(N * T)$ basic operations to choose parents. To update the external population, at most $O((\text{EN} + N) * tn)$ basic operations are needed. Furthermore, the update strategy of population also needs no more than $O(2N * N)$ basic operations. Altogether, the computational complexity of SHEA is $O(N * n) + O(N * T) + O((\text{EN} + N) * n) + O(2N * N) = O(N^2)$.

3. Experimental Study

3.1. Experimental Settings. In this section, the proposed algorithm is compared with four state-of-the-art algorithms such as NSGAIII [16], MOEA/DD [24], KnEA [25], and RVEA [26] on fifteen many-objective benchmark functions (MaF) from CEC2018 MaOP competition [27]. Each problem is tested for 5, 10, 15 objectives. NSGAIII [16] supplies and updates well-spread reference points adaptively to maintain the diversity among population members. MOEA/DD [24] exploits the merits of both dominance- and

Table 1: The mean and standard deviation values of IGD obtained by SHEA, NSGAIII, MOEA/DD, KnEA, and RVEA while "+," "=," "−" mean SHEA is better than, the same as, and worse than the compared algorithms.

Problem	SHEA	NSGAIII	MOEA/DD	KnEA	RVEA
MaF1-5	**1.1614e − 1 (1.32e − 3)**	2.0830e − 1 (1.00e − 2) +	3.0206e − 1 (1.07e − 2) +	1.3136e − 1 (1.91e − 3) =	3.2908e − 1 (6.21e − 2) +
MaF2-5	**9.6828e − 2 (2.82e − 3)**	1.3012e − 1 (2.56e − 3) =	1.3665e − 1 (3.72e − 3) =	1.3734e − 1 (3.62e − 3) =	1.2736e − 1 (1.42e − 3) =
MaF3-5	**6.5417e − 2 (2.81e − 3)**	9.7048e − 2 (1.52e − 3) =	1.1693e − 1 (1.75e − 3) +	1.6971e − 1 (9.55e − 2) +	8.0906e − 2 (6.70e − 3) =
MaF4-5	**1.8707e + 0 (4.09e − 2)**	3.2220e + 0 (5.56e − 1) +	7.7080e + 0 (2.14e − 1) +	2.9005e + 0 (2.52 − 1) +	4.8145e + 0 (1.30e − 0) +
MaF5-5	**1.8660e + 0 (3.63e − 2)**	2.5845e + 0 (1.15e + 0) +	6.1823e + 0 (1.00e + 0) +	2.6408e + 0 (8.04e − 2) +	2.3218e + 0 (3.07e − 1) +
MaF6-5	**3.9514e - 3 (4.97e - 4)**	4.9054e - 2 (9.12e − 3) +	7.6075e − 2 (4.42e − 3) +	8.0721e − 3 (2.28e − 3) =	9.6534e − 2 (3.42e − 2) +
MaF7-5	**2.7153e − 1 (9.43e − 3)**	3.4413e − 1 (1.20e − 2) +	1.7891e + 0 (8.07e − 1) +	3.2757e − 1 (8.09e − 3) +	4.4844e − 1 (1.10e − 3) +
MaF8-5	**8.9494e − 2 (2.04e − 3)**	2.1547e − 1 (2.19e − 2) +	3.3063e − 1 (3.61e − 2) +	2.9874e − 1 (8.77e − 2) +	4.8799e − 1 (8.73e − 2) +
MaF9-5	**9.3455e − 2 (3.24e − 3)**	6.5706e − 1 (1.32e − 1) +	2.5294e − 1 (1.34e − 2) +	5.6348e − 1 (1.82e − 1) +	3.6742e − 1 (7.00e − 2) +
MaF10-5	5.9695e − 1 (6.92e − 2)	**4.2888e − 1 (3.81e − 3)** −	5.4591e − 1 (3.64e − 2) −	5.1174e − 1 (7.85e − 3) −	4.3054e − 1 (4.90e − 3) −
MaF11-5	9.5446e − 1 (1.30e − 1)	4.6337e − 1 (1.69e − 3) −	5.7805e − 1 (9.69e − 3) −	5.7122e − 1 (2.10e − 2) −	**4.4446e − 1 (8.16e − 3)** -
MaF12-5	**1.0185e + 0 (1.27e − 2)**	1.1183e + 0 (4.00e − 3) +	1.2858e + 0 (1.43e − 2) +	1.2609e + 0 (1.70e − 2) +	1.1224e + 0 (2.51e − 3) +
MaF13-5	**1.0079e − 1 (3.64e − 3)**	2.9677e − 1 (5.22e − 2) +	2.4087e − 1 (2.54e − 2) +	2.2221e − 1 (2.01e − 2) +	6.6957e − 1 (1.36e − 1) +
MaF14-5	5.7391e − 1 (1.04e − 1)	7.9085e − 1 (3.74e − 1) +	**3.8193e − 1 (8.93e − 2)** −	7.6615e − 1 (2.77e − 1) +	7.1001e − 1 (2.02e − 1) +
MaF15-5	8.3177e − 1 (3.31e − 2)	1.0511e + 0 (4.49e − 2) +	**5.9102e − 1 (4.35e − 2)** −	3.4169e + 0 (1.84e + 0) +	6.1117e − 1 (4.50e − 2) −
MaF1-10	3.0459e − 1 (1.30e − 2)	3.1563e − 1 (7.02e − 3) =	4.8879e − 1 (4.58e − 2) +	**2.4052e − 1 (2.33e − 3)** −	6.6423e − 1 (8.71e − 2) +
MaF2-10	1.9990e − 1 (4.87e − 3)	2.3692e − 1 (2.48e − 2) +	2.9196e − 1 (7.15e − 2) +	**1.6521e − 1 (7.85e − 3)** =	4.8292e − 1 (1.82e − 1) +
MaF3-10	**8.3915e − 2 (2.62e − 3)**	9.3337e − 1 (3.12e + 0) +	1.1204e − 1 (1.03e − 3) =	1.1840e + 9 (5.06e + 9) +	9.8006e − 2 (5.33e − 3) =
MaF4-10	9.3360e + 1 (3.79e + 1)	1.2678e + 2 (6.34e + 0) +	4.3124e + 2 (1.96e + 1) +	**7.1006e + 1 (6.77e + 0)** −	2.3329e + 2 (5.10e + 1) +
MaF5-10	**6.0339e + 1 (6.65e + 0)**	1.1938e + 2 (2.75e − 1) +	2.9174e + 2 (7.70e + 0) +	8.1774e + 1 (5.26e + 0) +	1.0827e + 2 (1.73e + 1) +
MaF6-10	**3.9073e − 3 (3.35e − 4)**	2.1879e − 1 (4.79e − 2) +	1.2060e − 1 (8.53e − 3) +	8.4648e + 0 (7.78e + 0) +	3.3999e − 1 (2.44e − 1) +
MaF7-10	9.6508e − 1 (1.66e − 2)	1.1620e + 0 (7.18e − 2) +	2.7026e + 0 (3.72e − 1) +	**9.4855e − 1 (1.01e − 2) =**	2.4904e + 0 (3.56e − 1) +
MaF8-10	**1.3864e − 1 (3.50e − 3)**	4.6344e − 1 (6.16e − 2) +	9.1389e − 1 (2.39e − 2) +	2.6509e − 1 (3.86e − 2) +	1.0244e + 0 (2.14e − 1) +
MaF9-10	8.9161e − 1 (1.92e − 1)	9.5771e − 1 (3.22e − 1) +	**5.9743e − 1 (2.33e − 3)** −	8.5718e + 1 (6.92e + 1) +	1.1161e + 0 (2.69e − 1) +
MaF10-10	1.2266e + 0 (7.28e − 2)	**1.0965e + 0 (4.85e − 2)** −	1.2608e + 0 (2.88e − 2) =	1.2117e + 0 (5.05e − 2) =	1.1948e + 0 (1.04e − 1) =
MaF11-10	4.2556e + 0 (6.29e − 1)	**1.3422e + 0 (1.50e − 1)** -	1.4289e + 0 (1.06e − 2) −	1.3490e + 0 (4.51e − 2) −	1.3839e + 0 (4.88e − 2) −
MaF12-10	**4.3224e + 0 (5.40e − 2)**	5.1072e + 0 (1.42e − 1) +	6.0813e + 0 (2.79e − 1) +	5.2874e + 0 (6.12e − 2) +	4.8913e + 0 (4.87e − 2) +
MaF13-10	**1.6901e − 1 (1.21e − 2)**	4.1263e − 1 (6.56e − 2) +	4.4820e − 1 (3.29e − 2) +	2.1228e − 1 (2.37e − 2) +	9.3878e − 1 (2.89e − 1) +
MaF14-10	8.7548e − 1 (1.52e − 1)	1.4523e + 0 (5.34e − 1) +	**5.3428e − 1 (6.22e − 2)** −	1.7663e + 2 (2.00e + 2) +	6.6978e − 1 (5.91e − 2) −
MaF15-10	2.0029e + 0 (1.32e − 1)	3.0468e + 0 (4.38e + 0) +	**1.0133e + 0 (6.38e − 2)** −	1.9364e + 1 (8.54e + 0) +	1.0594e + 0 (4.47e − 2) −

Table 1: Continued.

Problem	SHEA	NSGAIII	MOEA/DD	KnEA	RVEA
MaF1-15	4.1181e − 1 (1.89e − 2)	**3.3510e − 1 (7.36e − 3)** −	6.3920e − 1 (4.13e − 2) +	3.3929e − 1 (3.83e − 2) −	7.3558e − 1 (5.35e − 2) +
MaF2-15	2.2440e − 1 (1.08e − 2)	2.4645e − 1 (2.39e − 2) =	3.1534e − 1 (3.10e − 2) +	**1.9241e − 1 (5.26e − 3)** =	7.8173e − 1 (8.16e − 2) +
MaF3-15	1.0297e − 1 (2.22e − 2)	1.7607e + 0 (4.79e + 0) +	1.1719e − 1 (1.28e − 3) =	2.2295e + 9 (4.30e + 9) +	**9.6650e − 2 (7.32e − 3)** =
MaF4-15	1.8657e + 4 (6.70e + 4)	4.5055e + 3 (4.36e + 2) −	1.5432e + 4 (2.45e + 3) −	**1.7224e + 3 (2.01e + 2)** −	7.7257e + 3 (1.96e + 3) −
MaF5-15	2.2259e + 3 (4.00e + 2)	3.1331e + 3 (3.77e + 1) +	7.3038e + 3 (6.71e + 1) +	**2.0495e + 3 (8.e + 1)** −	3.2985e + 3 (2.82e + 2) +
MaF6-15	**4.5840e − 3 (7.29e − 4)**	3.7141e − 1 (1.41e − 1) +	1.6153e − 1 (3.43e − 3) +	4.8429e + 1 (9.21e + 0) +	1.9771e − 1 (1.17e − 1) +
MaF7-15	**1.7351e + 0 (4.48e − 2)**	7.7037e + 0 (9.37e − 1) +	3.3764e + 0 (7.80e − 2) +	2.4545e + 0 (2.36e − 1) +	4.3845e + 0 (1.58e + 0) +
MaF8-15	**1.6721e − 1 (2.97e − 3)**	4.0978e − 1 (4.32e − 2) +	1.5460e + 0 (2.82e − 2) +	1.9336e − 1 (8.95e − 3)=	1.1960e + 0 (1.92e − 1) +
MaF9-15	**2.0922e − 1 (1.44e − 2)**	2.2160e + 0 (4.13e + 0) +	1.3164e + 0 (2.41e + 0) +	5.2669e − 1 (4.32e − 1) +	1.6125e + 0 (3.85e − 1) +
MaF10-15	1.7186e + 0 (6.28e − 2)	1.6348e + 0 (8.58e − 2) −	1.9986e + 0 (4.06e − 2) +	**1.6232e + 0 (5.10e − 2)** −	1.7417e + 0 (9.08e − 2) +
MaF11-15	8.8230e + 0 (1.22e + 0)	1.8586e + 0 (8.91e − 2) −	2.1955e + 0 (6.49e − 3) −	**1.7816e + 0 (6.20e − 2)** −	1.9469e + 0 (8.30e − 2) −
MaF12-15	8.4176e + 0 (1.66e − 1)	8.8410e + 0 (9.87e − 2) +	1.1457e + 1 (3.64e − 1) +	**7.3382e + 0 (1.38e − 1)** −	9.1349e + 0 (6.05e − 2) +
MaF13-15	1.8693e − 1 (1.01e − 2)	3.8155e − 1 (1.01e − 1) +	6.1376e − 1 (1.12e − 1) +	**1.5808e − 1 (1.46e − 2)** =	1.3296e + 0 (4.18e − 1) +
MaF14-15	2.1776e + 0 (9.69e − 1)	1.4120e + 0 (7.14e − 1) −	**4.4791e − 1 (7.87e − 2)** −	4.2956e + 1 (4.09e + 1) +	7.8507e − 1 (2.14e − 1) −
MaF15-15	4.3682e + 0 (9.00e − 1)	7.6508e + 0 (1.22e + 1) +	1.1636e + 0 (4.10e − 2) −	1.4233e + 2 (2.31e + 1) +	**1.1431e + 0 (3.90e − 2)**
+/−/=	—	32/9/4	29/12/4	25/11/9	30/10/5

The bold values indicate best performance.

decomposition-based approaches to balance the convergence and diversity of the evolutionary process. KnEA [25] is a knee point-driven EA to solve MaOPs. RVEA [26] adopts a scalarization approach named angle-penalized distance to balance convergence and diversity.

All of these fifteen test problems for each algorithm mentioned above are run on PlatEMO [28], and average data is given over 20 independent runs. In the proposed algorithm, the size of external population EN is about 2N; crossover probability of SBX operator is 1; the size of T is 0.1 N; J is 0.9; CR is 0.5, and F is 0.5. And other settings are the same as the standard of CEC2018 MaOP competition [27]. The algorithms are run on a PC with Intel Core i5-3210M (2.50 GHz for a single core and Windows 7 operating system) by using MATLAB language.

3.2. Performance Metrics. Comparison experiment employs inverted generational distance (IGD) [29] to judge the performances of these algorithms:

$$\mathrm{IGD}\left(P, P^{*}\right) = \frac{\sum_{x^{*} \in P^{*}} d\left(x^{*}, P\right)}{\left|P^{*}\right|}, \tag{7}$$

where P^{*} is a set of points uniformly sampled over the true PF, P is the population obtained from MOEAs, and $d(x^{*}, P)$is the Euclidean distance between x^{*} and its nearest neighbor in P.

IGD can comprehensively measure the convergence and diversity of population, and when IGD value is smaller, the population is closer to Pareto fronts. For each problem, around 10000 points on Pareto fronts are uniformly sampled to calculate IGD. Besides, in the sense of statistics comparison experiment, use Wilcoxon rank-sum test [30] whose significance level is set 0.05 to compare algorithms' mean IGD.

3.3. Comparative Studies. Table 1 shows the mean and standard deviation of IGDs which are given by the five MaOEAs on 5, 10, 15 objectives test problems over 20 independent runs. And the best performance for each test problem is marked in bold while "+," "=," "-" mean the proposed algorithm is better than, the same as, and worse than the compared algorithms.

On all forty-five problems in Table 1, SHEA statistically outperforms the compared algorithms on 21 problems, which reveals the good performance of the proposed algorithm in the form of IGD. NSGAIII, MOEA/DD, KnEA, and RVEA have better behavior than SHEA respectively on four, four, nine, five problems and SHEA does better than NSGAIII, MOEA/DD, KnEA and RVEA on thirty-two, twenty-nine, twenty-five, and thirty problems.

In fifteen MaFs, there are 8 problems (F1, F2, F4, F5, F7, F8, F9, and F15) that have partial PFs. The PF projections of these problems do not fully cover the unit hyperplane. The mean IGDs of SHEA are smaller than those of NSGAIII, MOEA/DD, KnEA, and RVEA on twenty-two, nineteen, sixteen, and twenty problems, separately. And for 6 problems (F3, F10, F11, F12, F13, and F14) with PF projection fully covering the unit hyperplane, there are respectively eleven, eleven, twelve, twelve problems that the mean IGDs of SHEA are smaller than those of NSGAIII, MOEA/DD, KnEA, and RVEA. As for the problem F6 whose PF is degraded, SHEA is superior to NSGAIII, MOEA/DD, KnEA, and RVEA on three problems in the form of IGD. All of these comparison results mentioned above indicate the best overall performance of SHEA on most problems and prove the excellent performance of simplified hypervolume in estimating convergence and diversity.

4. Conclusions

To simplify the calculation of hypervolume, a new simplified hypervolume is proposed to roughly estimate the convergence and diversity of solutions; then the new method is used in the selection operator and the update strategy of external population. And the proposed algorithm indicates good performance according to comparing experimental results with four state-of-the-art algorithms.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61806120, 61502290, 61401263, 61672334, and 61673251), China Postdoctoral Science Foundation (no. 2015M582606), Industrial Research Project of Science and Technology in Shaanxi Province (nos. 2015GY016 and 2017JQ6063), Fundamental Research Fund for the Central Universities (no. GK202003071), and Natural Science Basic Research Plan in Shaanxi Province of China (no. 2016JQ6045).

References

[1] G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer US, Boston, MA, USA, 2007.

[2] 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.

[3] N. Dong and C. Dai, “An improvement decomposition-based multi-objective evolutionary algorithm using multi-search strategy,” *Knowledge-Based Systems*, vol. 163, pp. 572–580, 2019.

[4] E. Zitzler and K. Simon, “Indicator-based selection in multiobjective search,” in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, Germany, pp. 832–842, September 2004.

[5] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[6] E. J. Hughes, “Evolutionary many-objective optimisation: many once or one many?” in *Proceedings of 2005 IEEE Congress on Evolutionary Computation*, pp. 222–227, Edinburgh, UK, September 2005.

[7] C. Dai, Y. Wang, and M. Ye, “A new evolutionary algorithm based on contraction method for many-objective optimization problems,” *Applied Mathematics and Computation*, vol. 245, pp. 191–205, 2014.

[8] X. F. Zou, Y. Chen, M. Z. Liu, and L. S. Kang, “A new evolutionary algorithm for solving many-objective optimization problems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 38, no. 5, pp. 1402–1412, 2008.

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

[10] Q. F. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposi-tion,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[11] S. Jiang and S. Yang, “An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts,” *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016.

[12] M. Elarbi, S. Bechikh, A. Gupta, L. Ben Said, and Y.-S. Ong, “A new decomposition-based NSGA-II for many-objective optimization,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 7, pp. 1191–1210, 2018.

[13] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, “Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 475–480, 2016.

[14] H.-L. Liu, F. Gu, and Q. Zhang, “Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[15] M. Asafuddoula, T. Ray, and R. Sarker, “A decomposition-based evolutionary algorithm for many objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2015.

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

[17] M. Asafuddoula, H. K. Singh, and T. Ray, “An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors,” *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2017.

[18] H. Lin, L. Chen, Q. Zhang, and K. Deb, “Adaptively allocating search effort in challenging many-objective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018.

[19] 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 Foundations of Genetic Algorithm X*, pp. 87–102, Orlando, FL, USA, 2009.

[20] E. Zitzler and S. Knzli, “Indicator-based selection in multi-objective search,” *Lecture Notes in Computer Science*, vol. 8, pp. 832–842, 2004.

[21] 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.

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

[23] K. V. Price, R. M. Storn, and J. A. Lampinen, “Differential evolution-A practical approach to global optimization,” *Natural Computing*, vol. 141, no. 2, 2005.

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

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

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

[27] R. Cheng, M. Li, Y. Tian et al., “Benchmark functions for CEC’2018 competition on many objective optimization,” Tech. Rep.B15. 2TT, CERCIA, School of Computer Science, University of Birmingham Edgbaston, Birmingham, U.K, 2017.

[28] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, “PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum],” *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[29] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[30] S. Robert, J. Torrie, and D. Dickey, *Principles and Procedures of Statistics: A Biometrical Approach*, McGraw-Hill, New York, NY, USA, 1997.