

Article

An Improved Multiobjective Particle Swarm Optimization Based on Culture Algorithms

Chunhua Jia * and Hong Zhu

School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611371, China; zhuhong@uestc.edu.cn

* Correspondence: jiachunhua624@163.com; Tel.: +86-133-8816-8253

Academic Editor: Javier Del Ser Lorente

Received: 14 February 2017; Accepted: 18 April 2017; Published: 25 April 2017

Abstract: In this paper, we propose a new approach to raise the performance of multiobjective particle swarm optimization. The personal guide and global guide are updated using three kinds of knowledge extracted from the population based on cultural algorithms. An epsilon domination criterion has been employed to enhance the convergence and diversity of the approximate Pareto front. Moreover, a simple polynomial mutation operator has been applied to both the population and the non-dominated archive. Experiments on two series of bench test suites have shown the effectiveness of the proposed approach. A comparison with several other algorithms that are considered good representatives of particle swarm optimization solutions has also been conducted, in order to verify the competitive performance of the proposed algorithm in solve multiobjective optimization problems.

Keywords: particle swarm optimization; cultural algorithm; multiobjective optimization

1. Introduction

Since many real-life problems are in fact multiobjective optimization problems (MOPs), evolutionary algorithms used to solve MOPs have attracted much attention in recent years. Multiobjective optimization evolutionary algorithms (MOEAs) mainly include two branches: dominance-based and decomposition-based [1]. Nondominated sorting genetic algorithm II (NSGA-II) [2], strength Pareto evolutionary algorithm 2 (SPEA2) [3] and multiobjective particle swarm optimization (MOPSO) [4,5] are in the first branch; multiobjective evolutionary algorithm based on decomposition (MOEA/D) [6,7] is in the second branch. For those dominance-based algorithms, the main goals of multiobjective optimization are that the non-dominated set obtained by an algorithm is a good approximation to the true Pareto front, and that this non-dominated set is uniformly distributed on the Pareto front [8].

Particle swarm optimization (PSO) is an evolutionary computation technique based on the social behavior of species, such as a flock of birds or a school of fish.

To handle MOPs, two problems need to be taken into account. The determination of personal and global guides is the first, and second is to maintain the diversity of the population.

Some efforts have been made in selecting the global and local guides. Carlos introduced the secondary repository that is based on Pareto optimality to store the non-dominated particles in MOPSO, and the global guide is selected using the adaptive grid [4]. In crowding distance (CD)-MOPSO, the global guide is selected using crowding distance in MOPSO [9]. In clustering MOPSO, the global guide is selected based on the clustering technique [10]. Mostaghim and Teich [11] selected the local guide based on the sigma method, while Branke and Mostaghim [12] compared the results of different methods in selecting the local guide.

Coello proposed the use of a cultural algorithm to deal with multiobjective optimization problems, in which the knowledge based on the cultural algorithm will affect the selection of both the global guide

and the local guide [13]. Cultural multiobjective quantum particle swarm optimization (MOQPSO) [1], Cultural MOPSO (CMOPSO) [14], constrained CMOPSO [15], and Cultural dynamic PSO (DPSO) [16] are all based on a cultural evolution mechanism.

A special domination principle and genetic operator have been used to raise the diversity of the Pareto front. Laumanns defined ϵ -box dominance to combine convergence and diversity [17], while Zitzler introduced the elitism mechanism that performs crossover and mutation on individuals selected from the union of population and repository [18]. Simulated Binary Crossover (SBX) has been used in the work of Tang [19] and Yu [20]. The Borg MOEA represents a class of algorithms whose operators are adaptively selected based on the problem [21]. Diversity-loss-based selection method is used in [22]. Bare-bones (BB)-MOPSO use an approach based on particle diversity to update the global particle leaders [23]. Pareto entropy MOPSO (peMOPSO) using cell distance based individual density to select the global guide [24].

Although there has been considerable research conducted on PSO proposed to solve MOPs, to raise the convergence and diversity of the approximate Pareto front further still remains an issue that needs to be considered.

An improved multiobjective particle swarm optimization has been developed in this paper. Situational, normative and topographical knowledge are to be used to update the personal and global guides. A polynomial mutation operator is applied to every particle in the swarm and in the repository to enhance the convergence and diversity of the Pareto front. Epsilon domination principle depends both on the situation in the adaptive grid and the objective value that has been used.

The paper is structured as follows. The concept of multiobjective problems, cultural algorithms, and particle swarm optimization are briefly reviewed in the next section. In Section 3, a new multiobjective particle swarm optimization is proposed to improve the convergence and spread to the true Pareto front. In Section 4, the results of the experiments and analysis are shown. Conclusions are given in Section 5.

2. Preliminary

2.1. Cultural Algorithms

A cultural algorithm usually includes five kinds of knowledge: situational, normative, topographical, domain, and historical knowledge [25].

Situational knowledge is a set of useful instances for the interpretation of the experiences of all individuals. Situational knowledge guides individuals to move toward exemplars.

Normative knowledge consists of a set of promising ranges of variables. It provides guidelines for individual adjustments. Normative knowledge leads individuals to plunge into a good range.

Topographical knowledge divides the whole functional landscape into cells representing different spatial characteristics, and each cell records the best individual in its specific ranges. Topographical knowledge leads individuals to catch up to the best cell.

Domain knowledge adopts information about the problem domain to lead the search. Domain knowledge is useful during the search process.

Historical knowledge keeps track of the history of the search process and records key events in the search. It might be either a considerable move in the search space or a discovery of a landscape change. Individuals use historical knowledge for guidance in selecting a direction in which to move.

Particles are evaluated by a performance function in each generation. An acceptance function determines which individuals will be used to update the belief space. Experiences of those chosen individuals are then added to the belief space by the update function. After that, knowledge from the belief space influences the selection of individuals for the next generation through the influence function. The framework of a cultural algorithm is shown in Figure 1.

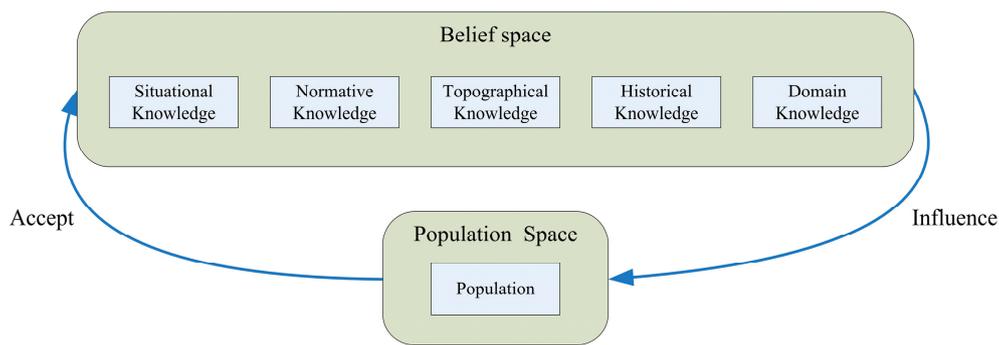


Figure 1. The framework of a cultural algorithm.

2.2. Multiobjective Optimization Problems

Multiobjective problems (MOPs) are problems whose goal is to optimize multiple objective functions simultaneously. A MOP global minimum problem with n objectives is defined as:

$$\begin{aligned} \min f(x) &= [f_1(x), f_2(x), \dots, f_n(x)] \\ \text{s.t. } g(x) &\leq 0, h(x) = 0 \end{aligned} \tag{1}$$

where $g(x) \leq 0$ and $h(x) = 0$ represent the constraints.

Definition 1. (Pareto dominance) [26]: x is said to dominate y (denoted as $x < y$) if $\forall i, f_i(x) \leq f_i(y), \exists i, f_i(x) < f_i(y)$.

Definition 2. (Pareto optimal set) [26]: $P^* = \{x | \neg \exists y \in \Omega, f(y) < f(x)\}$.

Definition 3. (Pareto front) [26]: $PF^* = \{f(x) = [f_1(x), f_2(x), \dots, f_n(x)] | x \in P^*\}$.

Definition 4. (ϵ -Pareto dominance) [17]: x is said to ϵ -dominate y , denoted as $x \prec_\epsilon y$, if $f(x) \cdot (1 + \epsilon) \prec f(y)$.

2.3. Particle Swam Optimization

The standard PSO [27] is:

$$\begin{aligned} v_{id}^{(k+1)} &= \omega \cdot v_{id}^{(k)} + c_1 \cdot r_1 (p_{id}^{(k)} - x_{id}^{(k)}) + c_2 \cdot r_2 (p_{gd}^{(k)} - x_{id}^{(k)}) \\ x_{id}^{(k+1)} &= x_{id}^{(k)} + v_{id}^{(k+1)} \end{aligned} \tag{2}$$

where $v_{id}^{(k)}$ and $x_{id}^{(k)}$ are the velocity and position of the d -th dimension of the i -th particle in the k -th iteration, respectively, ω is inertia weight, c_1 and c_2 are the acceleration coefficients. Parameters r_1 and r_2 are uniformly distributed random numbers within the range $[0, 1]$. The parameters $p_{id}^{(k)}$ and $p_{gd}^{(k)}$ indicate the best position that the i -th particle has found so far, and the best position that the global swarm has found so far. The procedure of PSO is shown in Algorithm 1.

Algorithm 1. The Procedure of Particle Swarm Optimization (PSO)

- (1) Calculate the fitness function of each particle.
 - (2) Update $p_{id}^{(k)}$ and $p_{gd}^{(k)}$.
 - (3) Update the velocity and position of each particle.
 - (4) If the stop criterion is not met, go to (1), else $p_g^{(k)}$ is the best position
-

3. The Proposed Algorithm

3.1. Three Kinds of Knowledge Based on Population

Situational knowledge is determined by all of the personal guide positions. Figure 2 gives the situational knowledge of a space with two objectives. The obtained local area is split into n parts identically in each dimension. Situational knowledge is used to generate the personal guide for each particle.

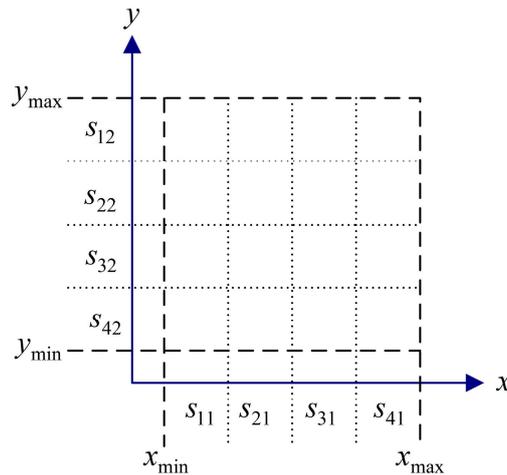


Figure 2. Situational knowledge.

Normative knowledge is based on the minimum and maximum values of non-dominated particles. Normative knowledge of a space with two objectives is shown in Figure 3. The normative knowledge in Figure 3 is updated from L_1, U_1, L_2, U_2 to L_1', U_1', L_2', U_2' .

$$L_i = \min f_i(x), U_i = \max f_i(x) \tag{3}$$

The normative space is later used to obtain the global guide of the MOPSO by establishing the topographical knowledge.

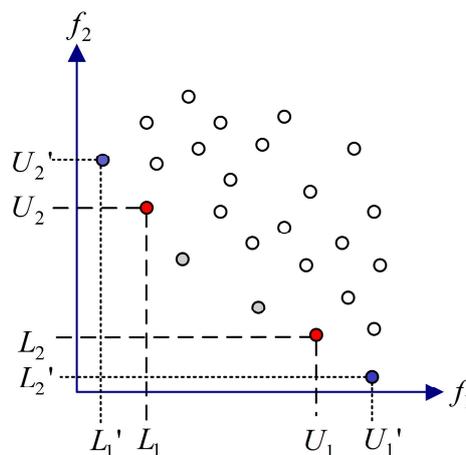


Figure 3. Normative knowledge.

In order to construct the topographical knowledge, the normative knowledge is employed and then the space is divided into grids, in which each of the resultant hypercubes will then be characterized

by all of the lower and upper limits of the hypercubes, as well as the number of non-dominated individuals that are located on that hypercube.

Figure 4 shows the topographical knowledge of a two-objective space. With each generation, the topographic knowledge will be updated. Topographical knowledge is used to find the global guide.

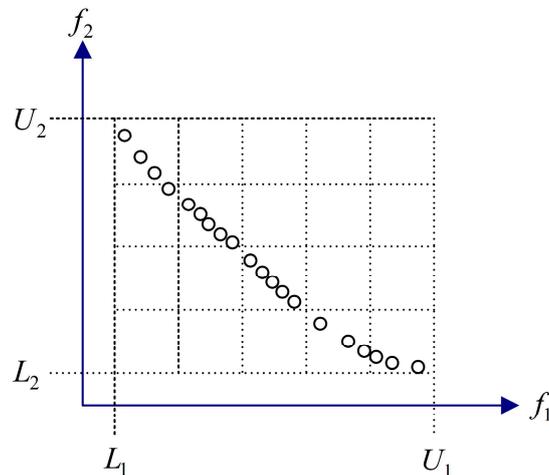


Figure 4. Topographical knowledge.

3.2. Personal Guide

As in cultural MOQPSO [1], one dimension of the position of one particle is substituted with a random number s generated in the band, decided by the personal guide of whole population. If the new one dominates the old one, then the new one is employed; otherwise, the old one continues to be used. The detailed procedure of a personal guide is shown in Algorithm 2.

Algorithm 2. The Detailed Procedure of A Personal Guide

For the d -th dimension of the i -th particle,

- (1) $b_{n+1} = \max(\{pbest_{id} \mid i = 1, 2, \dots, n\})$;
 $b_1 = \min(\{pbest_{id} \mid i = 1, 2, \dots, n\})$;
 $b_i = b_1 + (b_{n+1} - b_1) \cdot (i - 1) / n, i = 2, 3, \dots, n$

- (2) $s_i = \text{rand}(b_i, b_{i+1})$;

- (3) Replace x_i^d with s_i and get a new particle

- (4) If the newly generated particle dominates x_i , then it replaces x_i ; break;
 if not, the newly generated particle is discarded
-

3.3. Global Guide

Making use of topographical knowledge, a hypercube is selected based on Roulette Wheel Selection, then one member of this hypercube is randomly selected to be the global guide. The probability of selecting one hypercube is inversely proportional to the exponential of the number of non-dominated particles in that hypercube, and the exponential parameter is used here to increase the probability of selecting the hypercube that has less non-dominated particles in it.

3.4. Dominate Criteria Based on the Hypercube

Inferred from [17], a new domination criterion is proposed. The location of each pair of members in the non-dominated archive is used together with the objective value to determine the dominance. The hypercubes that are dominated by hypercube 3, 2 are shown in gray in Figure 5.

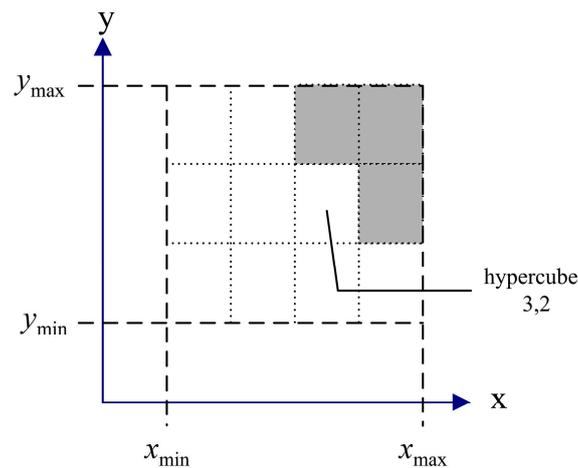


Figure 5. Hypercube.

3.5. Mutation Operator

A polynomial mutation is applied to each particle, and thereafter the non-dominated items produced are added into the repository. A polynomial mutation is applied to every member of the repository, the produced items are added into the repository, and then the repository based on dominates is updated. When the resulting offspring is non-dominated by the individuals in the repository, it is added into the repository. If any individual in the repository dominates the offspring, then the offspring is discarded.

The polynomial mutation is stated as [28]:

$$s_m = s + \alpha \cdot \beta_j \tag{4}$$

where $\alpha = \begin{cases} (2v)^{1/(p+1)} - 1, v < 0.5 \\ 1 - (2(1-v))^{1/(p+1)}, \text{ otherwise} \end{cases}$, $\beta = u_j - l_j$, p is a positive real number, v is a uniformly distributed random number in $[0, 1]$, and β_j is the range of j -th dimension.

3.6. Updating the Repository

Inferred from [29], in which the archive is updated based on the convergence and diversity metric, the archive is updated based on crowding distance in this paper. When the number of non-dominated solutions exceeds the limit size of the repository, first the crowding distance of every item in the archive is computed, and then the one that has the minimum crowding distance is deleted. This is repeated until the number of non-dominated solutions is equal to the limit size of the repository. The update function of the dominance archive is shown in Algorithm 3.

Algorithm 3. The Update Function of the Dominance Archive

Suppose element f' is a new element that is to be judged whether to enter the repository (rep).

- (1) Input rep, f
 - (2) If $\neg \exists f' \in \text{rep} : \text{hypercube}(f) \prec \text{hypercube}(f')$ then $\text{rep} = \text{rep} \cup \{f\}$
 - (3) If $\exists f' : \text{hypercube}(f') = \text{hypercube}(f) \cap f \prec \epsilon + f'$ then $\text{rep} = \text{rep} \cup \{f\} f'$
 - (4) If $\neg \exists f' : \text{hypercube}(f') = \text{hypercube}(f) \cup \text{hypercube}(f') \prec \text{hypercube}(f)$, then $\text{rep} = \text{rep} \cup \{f\}$
 - (5) Output rep
-

4. Experiments

The proposed approach is compared to the following several multiobjective evolutionary algorithms to test its performance: MOPSO [4], cultural MOPSO [14], cultural quantum-behaved

MOPSO [1], NSGA-II [2], and crowding distance-based MOPSO [9]. The implementation of some algorithms is available at <http://delta.cs.cinvestav.mx/~ccoello/EMOO/EMOOsoftware.html>.

ZDT and DTLZ test functions were taken to verify our approach [26]. The number of objectives in the ZDT test function is 2 and the number of objectives in the DTLZ test function is 3. The number of variables is 30 for ZDT1-3, 10 for ZDT4, 10 for ZDT6, 10 for DTLZ1-6 and 20 for DTLZ7.

The algorithm is run for a population size of 100, a repository size of 100 particles, and 30 divisions for the adaptive grid in each dimension of the objective space. The number of fitness function evaluations is 30,000. The results that obtained over 30 independent runs are reported in the following. The parameter of the PSO is time-varied as below [30,31]:

$$\omega = 0.9 - (\text{ita}/\text{max_ita}) \times (0.9 - 0.4) \tag{5}$$

$$c_1 = 2.5 - (\text{ita}/\text{max_ita}) \times (2.5 - 0.5) \tag{6}$$

$$c_2 = 0.5 - (\text{ita}/\text{max_ita}) \times (0.9 - 0.4) \tag{7}$$

4.1. Quantitative Performance Evaluations

Hypervolume is adopted as a quantitative measure in this article, as it is a measure of the performance in both convergence and diversity [32]. Hypervolume is defined mathematically as follows:

$$Hv = \hat{\gamma} \left(\bigcup_{p \in PF_i} \{x | p < x < r\} \right) \tag{8}$$

where $\hat{\gamma}$ is the Lebesgue measure, and $r \in Ri$ is a reference vector that is dominated by all of the valid candidate solutions in PF_i .

Table 1 shows the mean and standard deviation values of hypervolume for over 30 independent runs obtained by all of the six algorithms. For further analysis, the T test is performed to evaluate the significant difference between two independent samples, and the results are illustrated in Table 1, where the value of α is set to 0.05. The best result of one test function with different methods is shown in bold. It can be seen that the proposed improved cultural MOPSO (ICMOPSO) has achieved the best performance in hypervolume for nine of the 12 test functions, and it is superior to MOPSO for all 12 of the test functions, superior to Cultural MOPSO except in ZDT6, superior to NSGA-II except in DTLZ5, superior to CD-MOPSO except in five functions, and finally inferior to CD-MOPSO only in DTLZ2.

Table 1. Hypervolume values for 30 runs of the test functions.

| Test Functions | | MOPSO | IC MOPSO | Cultural MOQPSO | Cultural MOPSO | NSGA-II | CD MOPSO |
|----------------|--------------------------|-------------------------|-----------------------|-----------------------|-------------------------|-------------------------|-------------------------|
| ZDT1 | mean | 5.56×10^{-1} | 8.73×10^{-1} | 8.42×10^{-1} | 8.66×10^{-1} | 8.11×10^{-1} | 8.74×10^{-1} |
| | standard deviation (std) | 9.03×10^{-2} | 5.38×10^{-3} | 8.01×10^{-2} | 7.00×10^{-3} | 8.46×10^{-3} | 4.76×10^{-3} |
| | p-value | 8.42×10^{-2} | | 3.72×10^{-2} | 8.82×10^{-5} | 6.65×10^{-40} | 6.25×10^{-1} |
| ZDT2 | mean | 2.88×10^{-1} | 5.38×10^{-1} | 4.72×10^{-1} | 4.20×10^{-1} | 1.20×10^{-1} | 5.37×10^{-1} |
| | std | 1.19×10^{-1} | 7.88×10^{-3} | 1.07×10^{-1} | 1.91×10^{-1} | 2.22×10^{-2} | 6.43×10^{-3} |
| | p-value | 1.69×10^{-16} | | 1.40×10^{-3} | 1.31×10^{-5} | 6.53×10^{-66} | 8.25×10^{-1} |
| ZDT3 | mean | 1.00 | 1.01 | 9.39×10^{-1} | 7.61×10^{-1} | 7.60×10^{-1} | 1.01 |
| | std | 5.17×10^{-3} | 4.01×10^{-3} | 1.43×10^{-1} | 4.00×10^{-1} | 1.31×10^{-2} | 4.37×10^{-3} |
| | p-value | 1.72×10^{-7} | | 0.0110 | 1.35×10^{-3} | 2.35×10^{-66} | 8.76×10^{-1} |
| ZDT4 | mean | 0.00 | 8.71×10^{-1} | 8.67×10^{-1} | 0.00 | 3.66×10^{-1} | 0.00 |
| | std | 0.00 | 6.08×10^{-3} | 5.18×10^{-3} | 0.00 | 1.77×10^{-1} | 0.00 |
| | p-value | 1.80×10^{-118} | | 1.59×10^{-2} | 1.80×10^{-118} | 1.80×10^{-118} | 1.80×10^{-118} |
| ZDT6 | mean | 4.94×10^{-1} | 5.04×10^{-1} | 4.65×10^{-1} | 5.03×10^{-1} | 0.00 | 5.05×10^{-1} |
| | std | 7.43×10^{-3} | 4.88×10^{-3} | 1.12×10^{-1} | 7.63×10^{-3} | 0.00 | 5.77×10^{-3} |
| | p-value | 5.97×10^{-08} | | 6.31×10^{-2} | 8.61×10^{-1} | 3.40×10^{-110} | 8.76×10^{-1} |
| DTLZ1 | mean | 0.00 | 1.30 | 1.30 | 0.00 | 0.00 | 0.00 |
| | std | 0.00 | 2.02×10^{-3} | 3.07×10^{-3} | 0.00 | 0.00 | 0.00 |
| | p-value | 2.90×10^{-156} | | 6.44×10^{-1} | 2.90×10^{-156} | 2.90×10^{-156} | 2.90×10^{-156} |

Table 1. Cont.

| Test Functions | | MOPSO | IC MOPSO | Cultural MOQPSO | Cultural MOPSO | NSGA-II | CD MOPSO | | | | |
|----------------------|---------|-------------------------|-----------------------|------------------------|-------------------------|-------------------------|-------------------------|--------|--|-------|--|
| DTLZ2 | mean | 6.74×10^{-1} | 7.15×10^{-1} | 4.27×10^{-1} | 2.46×10^{-1} | 6.93×10^{-1} | 7.20×10^{-1} | | | | |
| | std | 1.50×10^{-2} | 9.74×10^{-3} | 2.41×10^{-2} | 1.26×10^{-2} | 8.35×10^{-3} | 8.12×10^{-3} | | | | |
| | p-value | 4.76×10^{-18} | | 3.83×10^{-54} | 1.33×10^{-78} | 3.17×10^{-13} | 3.85×10^{-2} | | | | |
| DTLZ3 | mean | 0.00 | 7.16×10^{-1} | 7.15×10^{-1} | 0.00 | 0.00 | 0.00 | | | | |
| | std | 0.00 | 7.63×10^{-3} | 9.61×10^{-3} | 0.00 | 0.00 | 0.00 | | | | |
| | p-value | 8.20×10^{-108} | | 9.47×10^{-1} | 8.20×10^{-108} | 8.20×10^{-108} | 8.20×10^{-108} | | | | |
| DTLZ4 | mean | 6.82×10^{-1} | 7.29×10^{-1} | 5.46×10^{-1} | 1.67×10^{-1} | 6.65×10^{-1} | 7.22×10^{-1} | | | | |
| | std | 1.56×10^{-2} | 7.45×10^{-3} | 5.70×10^{-2} | 6.76×10^{-2} | 3.11×10^{-2} | 8.74×10^{-3} | | | | |
| | p-value | 1.55×10^{-21} | | 9.53×10^{-25} | 6.11×10^{-47} | 9.22×10^{-16} | 8.66×10^{-4} | | | | |
| DTLZ5 | mean | 4.34×10^{-1} | 4.39×10^{-1} | 4.35×10^{-1} | 2.43×10^{-1} | 4.38×10^{-1} | 4.38×10^{-1} | | | | |
| | std | 6.21×10^{-3} | 4.94×10^{-3} | 8.00×10^{-3} | 4.96×10^{-3} | 5.74×10^{-3} | 5.86×10^{-3} | | | | |
| | p-value | 1.65×10^{-3} | | 3.72×10^{-2} | 2.13×10^{-77} | 5.64×10^{-1} | 3.14×10^{-1} | | | | |
| DTLZ6 | mean | 4.29×10^{-1} | 4.40×10^{-1} | 4.21×10^{-1} | 2.36×10^{-1} | 0.00 | 4.39×10^{-1} | | | | |
| | std | 6.99×10^{-3} | 5.97×10^{-3} | 7.53×10^{-3} | 2.81×10^{-2} | 0.00 | 6.85×10^{-3} | | | | |
| | p-value | 1.43×10^{-8} | | 5.64×10^{-15} | 3.58×10^{-43} | 9.89×10^{-102} | 6.28×10^{-1} | | | | |
| DTLZ7 | mean | 1.27 | 1.43 | 1.21 | 5.44×10^{-1} | 1.47×10^{-1} | 1.38 | | | | |
| | std | 2.93×10^{-2} | 3.20×10^{-2} | 4.10×10^{-2} | 8.36×10^{-2} | 4.43×10^{-2} | 4.25×10^{-2} | | | | |
| | p-value | 1.19×10^{-28} | | 2.91×10^{-31} | 1.98×10^{-51} | 5.57×10^{-73} | 5.54×10^{-6} | | | | |
| Better/similar/worse | | 12/0/0 | | 9/3/0 | | 11/1/0 | | 11/1/0 | | 5/6/1 | |

4.2. Comparison of the Results Using Hypercube-Based ϵ -Domination and Normal Domination Hypervolume

To demonstrate the effectiveness of the hypercube-based ϵ -domination criteria introduced in this paper, DTLZ1 and DTLZ7 are chosen as examples.

Figures 6 and 7 show the Pareto fronts obtained using two kinds of dominate criteria for DTLZ1 and DTLZ7, respectively. It can be observed that the approximate Pareto fronts found by the improved Cultural MOPSO using hypercube-based ϵ -domination are more well-distributed.

The adjacent members of the repository may be deleted based on the hypercube-based domination criterion, because they may locate in the same grid which is not expected in scenes like ZDT4, ZDT6 and DTLZ6. So, for ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ5, and DTLZ6, normal domination criterion should be used.

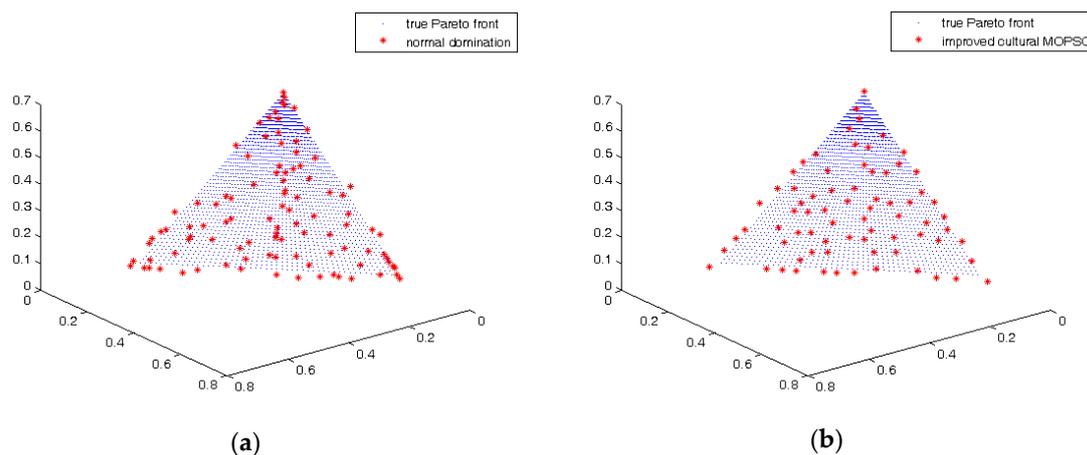


Figure 6. Pareto fronts for DTLZ1: (a) the Pareto fronts obtained using normal domination criterion; (b) the Pareto fronts obtained using hypercube-based ϵ -domination.

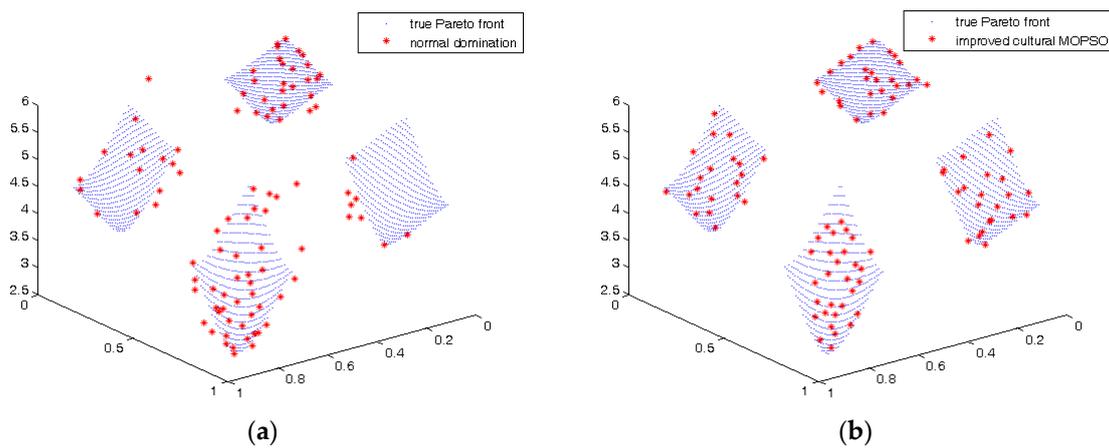


Figure 7. Pareto fronts for DTLZ7: (a) the Pareto fronts obtained using normal domination criterion; (b) the Pareto fronts obtained using hypercube-based ϵ -domination.

4.3. Comparison of the Result with and without the Mutation Operator

The comparison of the Pareto fronts of ZDT1 and ZDT3 with and without the mutation operator is shown in Figures 8 and 9. It can be seen that using the mutation operator can result in a more well-distributed approximation of the Pareto front.

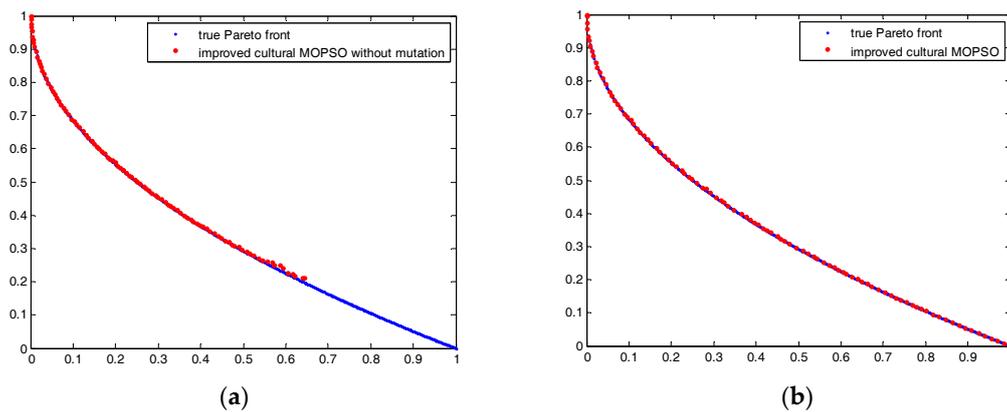


Figure 8. Pareto fronts for ZDT1: (a) the Pareto fronts obtained without using the mutation operator; (b) the Pareto fronts obtained using the mutation operator.

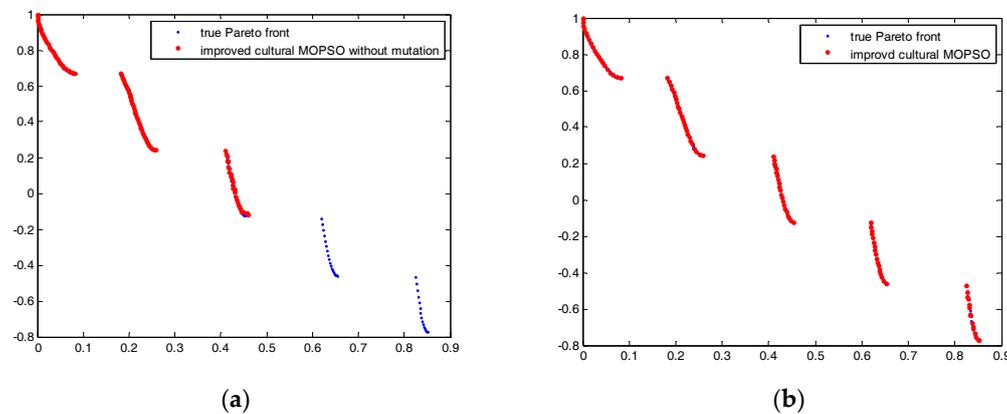


Figure 9. Pareto fronts for ZDT3: (a) the Pareto fronts obtained without using the mutation operator; (b) the Pareto fronts obtained using the mutation operator.

5. Conclusions

A new approach has been proposed in this paper to improve the performance of MOPSO. It updates the personal and global guides based on three kinds of knowledge that are extracted from a cultural algorithm. An epsilon domination criterion has been utilized. What is more, a polynomial mutation operator is applied in order to raise the performance of the Pareto front. Experiments on a series of ZDT and DTLZ test functions has been conducted to compare the proposed method with several state-of-the-art MOPSO algorithms. The results indicate that the proposed approach is a viable alternative, since its average performance is highly competitive. The performance of the proposed algorithm in dynamic optimization problems merits further study in future works.

Acknowledgments: The research work was supported by the National Defense Pre-Research Foundation of China under Grant No. 9140A27020215DZ02001.

Author Contributions: Chunhua Jia and Hong Zhu conceived and designed the experiments; Chunhua Jia performed the experiments; Chunhua Jia and Hong Zhu analyzed the data; Hong Zhu contributed analysis tools; Chunhua Jia wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, T.; Jiao, L.; Ma, W.; Ma, J.; Shang, R. A new quantum-Behaved particle swarm optimization based on cultural evolution mechanism for multiobjective problems. *Appl. Soft Comput.* **2016**, *46*, 267–283. [[CrossRef](#)]
2. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
3. Zitzler, E.; Laumanns, M.; Thiele, L. Spea2: Improving the strength pareto evolutionary algorithm. In Proceedings of the Evolution Methods Design Optimization Control Applications Industrial Problems, Zurich, Switzerland, 1 May 2002.
4. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 256–279. [[CrossRef](#)]
5. Coello, C.A.C.; Lechuga, M.S. Mopso: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 12 May 2002.
6. Zhang, Q.; Li, H. MOEA/D: A multi-Objective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [[CrossRef](#)]
7. Liu, H.; Gu, F.; Zhang, Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Trans. Evol. Comput.* **2014**, *18*, 450–455. [[CrossRef](#)]
8. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [[CrossRef](#)]
9. Raquel, C.R.; Naval, P.C. An effective use of crowding distance in multiobjective particle swarm optimization. In Proceedings of the Conference on Genetic and Evolutionary Computation, Washington, DC, USA, 25–29 June 2005.
10. Pulido, G.T.; Coello, C.A.C. Using clustering techniques to improve the performance of a particle swarm optimizer. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Seattle, WA, USA, 26–30 June 2004.
11. Mostaghim, S.; Teich, J. Strategies for finding good local guides in multi-Objective particle swarm optimization (MOPSO). In Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2003), Los Alamitos, CA, USA, 26 April 2003; IEEE Computer Society Press: Indianapolis, IN, USA, 2003. [[CrossRef](#)]
12. Branke, J.; Mostaghim, S. About Selecting the Personal Best in Multi-Objective Particle Swarm Optimization. In Proceedings of the Parallel Problem Solving From Nature (PPSN IX) International Conference, Reykjavik, Iceland, 9–13 September 2006.
13. Coello, C.A.C.; Becerra, R.L. Evolutionary multiobjective optimization using a cultural algorithm. In Proceedings of the Swarm Intelligence Symposium, Indianapolis, IN, USA, 24 April 2003.
14. Daneshyari, M.; Yen, G.G. Cultural-Based multiobjective particle swarm optimization. *IEEE Trans. Syst. Man Cybern.* **2011**, *41*, 553–567. [[CrossRef](#)] [[PubMed](#)]

15. Daneshyari, M.; Yen, G.G. Constrained Multiple-Swarm Particle Swarm Optimization within a Cultural Framework. *IEEE Trans. Syst. Man Cybern.* **2012**, *42*, 475–490. [[CrossRef](#)]
16. Daneshyari, M.; Yen, G.G. Cultural-Based particle swarm for dynamic optimisation problems. *Int. J. Syst. Sci.* **2011**, *43*, 1–21. [[CrossRef](#)]
17. Laumanns, M.; Thiele, L.; Deb, K. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evol. Comput.* **2002**, *10*, 263–282. [[CrossRef](#)] [[PubMed](#)]
18. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **2000**, *8*, 173–195. [[CrossRef](#)]
19. Tang, L.; Wang, X. A Hybrid Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems. *IEEE Trans. Evol. Comput.* **2013**, *17*, 20–45. [[CrossRef](#)]
20. Yu, G. Multi-Objective estimation of Estimation of Distribution Algorithm based on the Simulated binary Crossover. *J. Conver. Inf. Technol.* **2012**, *7*, 110–116.
21. Hadka, D.; Reed, P. Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework. *Evol. Comput.* **2013**, *21*, 231–259. [[CrossRef](#)] [[PubMed](#)]
22. Gee, S.B.; Arokiasami, W.A.; Jiang, J. Decomposition-Based multi-Objective evolutionary algorithm for vehicle routing problem with stochastic demands. *Soft Comput.* **2016**, *20*, 3443–3453. [[CrossRef](#)]
23. Zhang, Y.; Gong, D.W.; Ding, Z. A bare-Bones multi-Objective particle swarm optimization algorithm for environmental/economic dispatch. *Inf. Sci.* **2012**, *192*, 213–227. [[CrossRef](#)]
24. Hu, W.; Yen, G.G.; Zhang, X. Multiobjective particle swarm optimization based on Pareto entropy. *Ruan Jian Xue Bao/J. Softw.* **2014**, *25*, 1025–1050. (In Chinese).
25. Peng, B.; Reynolds, R.G. Cultural algorithms: Knowledge learning in dynamic environments. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 1751–1758.
26. Coello, C.A.C.; Veldhuizen, D.A.V.; Lamont, G.B. *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed.; Kluwer Academic Publishers: New York, NY, USA, 2007; pp. 1–19.
27. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
28. Li, L.M.; Lu, K.D.; Zeng, G.Q.; Wu, L.; Chen, M.-R. A novel real-Coded population-Based extremal optimization algorithm with polynomial mutation: A non-Parametric statistical study on continuous optimization problems. *Neurocomputing* **2016**, *174*, 577–587. [[CrossRef](#)]
29. Chen, Y.; Zou, X.; Xie, W. Convergence of multi-Objective evolutionary algorithms to a uniformly distributed representation of the Pareto front. *Inf. Sci.* **2011**, *181*, 3336–3355. [[CrossRef](#)]
30. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July 1999.
31. Ratnaweera, A.; Halgamuge, S.; Watson, H.C. Self-Organizing hierarchical particle swarm optimizer with time-Varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255. [[CrossRef](#)]
32. Zitzler, E.; Thiele, L. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Case Study. In Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN V), Amsterdam, The Netherlands, 27–30 September 1998; Agoston, E.E., Thomas, B., Marc, S., Schwefel, H.-P., Eds.; Springer: Berlin / Heidelberg, Germany, 1998.

