Generating multiple reference vectors for a class of many-objective optimization problems with degenerate Pareto fronts

Yicun Hua^{1,2} · Yaochu Jin^{1,2,3} · Kuangrong Hao^{1,2} · Yuan Cao⁴

Received: 11 January 2020 / Accepted: 3 March 2020 / Published online: 14 March 2020
© The Author(s) 2020

Abstract

Many-objective optimization problems with degenerate Pareto fronts are hard to solve for most existing many-objective evolutionary algorithms. This is particularly true when the shape of the degenerate Pareto front is very narrow, and there are many dominated solutions near the Pareto front. To solve this particular class of many-objective optimization problems, a new evolutionary algorithm is proposed in this paper. In this algorithm, a set of reference vectors is generated to locate the potential Pareto front and then generate a set of location vectors. With the help of the location vectors, the solutions near the Pareto front are mapped to the hyperplane and clustered to generate more reference vectors pointing to Pareto front. This way, the location vectors are able to efficiently guide the population to converge towards the Pareto front. The effectiveness of the proposed algorithm is examined on two typical test problems with degenerate Pareto fronts, namely DTLZ5 and DTLZ6 with 5–40 objectives. Our experimental results show that the proposed algorithm has a clear advantage in dealing with this class of many-objective optimization problems. In addition, the proposed algorithm has also been successfully applied to optimization of process parameters of polyester fiber filament melt-transportation.

Keywords Evolutionary algorithm · Many-objective optimization · Irregular Pareto front · Mapping · Hierarchical clustering · Polyester filament

Introduction

Many-objective optimization problems (MaOPs) refer to those multi-objective optimization problems with more than

three objectives [1,2]. Interestingly, most MaOPs in the real world are problems with irregular Pareto fronts [3,4]. The Pareto optimal solutions of these problems are distributed only in part of the objective space. Several characteristics make such problems more difficult to solve than problems with regular Pareto fronts [5,6]. Firstly, since the distribution of Pareto front is unknown, it is hard to identify the real Pareto front area. The dominated individuals affect the judgment and waste computing resources, thus slowing down convergence, and even misleading the convergence direction. Secondly, since the Pareto optimal solutions are only distributed in part of the objective space, it is difficult to randomly generate enough individuals in the Pareto area. Thirdly, the irregular shape of the Pareto front makes it more difficult to partition the objective space uniformly, especially in high-dimensional space, which brings difficulties to decomposition-based algorithms.

In recent years, MaOPs with irregular Pareto fronts have received increasing attention in the field of evolutionary optimization. Several evolutionary algorithms have been developed for this kind of problems. These algorithms can be roughly divided into two categories. Firstly, algorithms

✉ Yaochu Jin
yaochu.jin@surrey.ac.uk

Yicun Hua
huayicun@dh.u.edu.cn

Kuangrong Hao
krhao@dh.u.edu.cn

Yuan Cao
caoyuan@dh.u.edu.cn

¹ College of Information Science and Technology, Donghua University, Shanghai 201620, People's Republic of China

² Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, Shanghai, People's Republic of China

³ Department of Computer Science, University of Surrey, Guildford, Surrey GU2 7XH, UK

⁴ Human Resource Department, Donghua University, Shanghai 201620, People's Republic of China

based on clustering or space division. In our recent work [6], a clustering based multi-objective evolutionary algorithm called CA-MOEA was proposed, which adaptively generates a set of cluster centers in the current population as the reference points to maintain diversity and accelerate convergence for problems with irregular Pareto fronts. However, as CA-MOEA is mainly based on non-dominated sorting, it encounters resistance in dealing with many-objective problems. MaOEA/C [7] partitions the population by two clustering approaches and selects individuals with best convergence to form the final population. However, due to frequent changes in clustering results, population convergence will be affected. RdEA [8] predicts the geometry of the Pareto front according to the distribution of the population, and proposes a region division method to partition the objective space. In each region, individuals closer to the reference point or the center point of the region are selected. [9,10] are also algorithms based on division. Secondly, algorithms based on adjustment of reference vectors. RVEA* [11], a variant of RVEA [11], was proposed to deal with irregular Pareto fronts. In this algorithm, if no individual is associated with a reference vector, it will be replaced by a randomly generated vector. Consequently, the convergence directions varies with iteration and it is also difficult to guarantee an even partition of the population, resulting in wasted computing resources and deteriorated convergence performance. The very recently proposed iRVEA [12] suggested a new vector adaption method to further improve the performance of RVEA on irregular problems. A-NSGA-III [13] is a variant of NSGA-III [13] for irregular Pareto fronts, reference points with which no individual is associated will be deleted and relocated around a reference point associated with more than two individuals. The algorithm can easily be misled by individuals in incorrect directions at the beginning of iteration. Meanwhile, such repeated detection, deletion and relocation operations are inefficient and leads to a high computational complexity. Furthermore, the original reference points need to be kept for diversity considerations, requiring additional computation resources. Similar to other algorithms, relocating reference points may cause useless calculation and an uneven distribution of the solutions. There are several variants of MOEA/D [14] for problems with complex Pareto fronts. For example, MOEA/D-AWA [15] produces new weight vectors by non-dominated solutions found by now to adjust the weight vectors periodically for finding the appropriate convergence direction. Self-organizing map (SOM) is applied in MOEA/D-SOM [16] to adjusting the weight vectors for MOEA/D. The problem is that training a SOM network costs a lot of calculation and the performance of SOM depends on the distribution of initial random individuals which probably not reflecting the true distribution of the Pareto front. Besides, MaOEA/D-2ADV [17], MOEA/D-SAS [18], and MOEA/D-AM2M [19] all make efforts to get

reasonable results on problems with irregular Pareto fronts. CLIA [20] use incremental Support Vector Machine (SVM) to judge the effectiveness of reference vectors and adjust the distribution of reference vectors. Obviously, training classifier requires a lot of computation. Moreover, as dominated individuals usually do not reflect the distribution of Pareto fronts, the training results have a risk of error. Meanwhile, changing the reference vectors time to time will affect the speed of convergence.

Especially, if an M-objective optimization problem has a Pareto front of a dimension less than (M-1), the Pareto front is called a degenerate Pareto front [21] which is an important type of irregular Pareto front. In particular, DTLZ5 and DTLZ6 with more than three objectives are a class of degenerate problems that are especially hard for MOEAs to converge to the Pareto front. To the best of our knowledge, no existing algorithms are able to obtain well converged solutions on DTLZ5 and DTLZ6 problems with more than three objectives.

This work aims to develop an effective evolutionary algorithm tailored for solving a class of challenging MaOPs with degenerated Pareto fronts, such as DTLZ5 and DTLZ6 with more than three objectives. To this end, a many-objective evolutionary algorithm based on Pareto front location and piecewise mapping clustering (called LC-MaOEA) is proposed. In LC-MaOEA, the potential Pareto front regions are detected and a set of reference vectors based on clustering of projected solutions are produced within these regions to maintain stable and reliable convergence. To be specific, LC-MaOEA can be seen a decomposition-based evolutionary algorithm, in which the population is guided by four sets of reference vectors, namely, location vectors, axis vectors, mapping-based cluster center vectors and in some cases, Gaussian random vectors. These vectors cooperate together to maintain diversity and speed up convergence. Meanwhile, neighborhood-based mating selection mechanism is used to improve the ability of the algorithm to produce more individuals in Pareto front area.

The main contributions of this work can be summarized as follows:

- 1) A novel evolutionary algorithm LC-MaOEA is proposed. In this algorithm, a new vector-based Pareto front location method is designed, which improves the utilization of computing resources and makes the population converge to the Pareto front area more efficiently. The proposed algorithm can solve the DTLZ5 and DTLZ6 problems with more than three objectives very well, and has achieved good results in dealing with similar practical application problems.
- 2) A new method for generating reference vectors is proposed. Four kinds of vectors are generated by this method. Among them, location vectors locate the approximate

position of Pareto front, axis vectors are used to find the boundary individuals, and cluster center vectors are generated by clustering the individuals mapped to the piecewise hyperplanes to improve diversity. Finally, a set of Gaussian random vectors will be generated within the Pareto front area if the number of cluster center vectors is not enough.

- 3) The proposed LC-MaOEA is examined on DTLZ5 and DTLZ6 problems with 5 to 40 objectives and compared with four state-of-the-art evolutionary algorithms designed for irregular Pareto fronts. A widely used performance indicators are adopted to evaluate the performance of LC-MaOEA. Furthermore, LC-MaOEA is applied to optimize the polyester filament melt-transportation process. Our experimental results demonstrate that LC-MaOEA is promising for solving this kind of real world problems.

Basic ideas

Pareto front location

The Pareto front of DTLZ5 and DTLZ6 with more than three objectives is distributed in a narrow area of the objective space. Figure 1 shows the Pareto front of 3-objective DTLZ5 [22] problem which is a typical problem with degenerate Pareto front. The hollow circles are reference points produced by Das and Dennis's systematic approach [23], and the reference vectors are generated by the origin and these reference points. The solid points are Pareto optimal solutions of the 3-objective DTLZ5 problem, the stars are effective vectors with the smallest angle to at least one Pareto optimal solution. Figure 2 shows the dominated solutions of the 3-objective DTLZ5 problem. The hollow circles and the stars have the same meaning as in Fig. 1, but the solid points represent the dominated individuals in one of the early iteration. It can be seen from Fig. 1 that in a degenerate problem, a very small proportion of reference vectors are effective. The remaining vectors not only occupy computing resources, but also mislead the convergence direction. That is why algorithms using fixed reference vectors are difficult to perform well in such problems.

To remedy this, some algorithms adjust the reference vectors according to the distribution of current population. However, as shown in Fig. 2, since the distribution of dominated population do not accurately reflect the real distribution of Pareto front, those ineffective reference vectors will lead to an ineffective area, thus population cannot converge to the correct PF. Moreover, since population changes from generation to generation, the frequently changing reference vectors affects convergence rate.

Fig. 1 Pareto front of 3-objective DTLZ5 problem

Fig. 2 Dominated solutions of 3-objective DTLZ5 problem

Based on the above considerations, in this paper, we proposed a vector-based detection method to locate the Pareto front and providing a reliable convergence direction for the population. Firstly, we set a number of uniformly distributed reference vectors by Das and Dennis's systematic approach [23]. Note that these reference vectors are not only set for detection, but also working as the convergence direction in the early iteration. Meanwhile, every individual will be assigned to a vector which has the minimum angle with it, and those vectors that have no less than one individual assigned to them is remarked as possible effective vectors. After that, we evaluate the change of possible effective vectors, and if the distribution of possible effective vectors tends to be stable, we get the final set of effective vectors which will be treated as location vectors. The location vectors will keep guiding the population until the end of the iteration. This

method clarifies the convergence direction and speeds up the convergence rate.

Piecewise mapping-based cluster center vectors

Decomposition-based evolutionary algorithm has been regarded as a promising approach for handling MaOPs. These algorithms decompose a problem into a number of sub-problems by a set of vectors. In this paper, we decided to apply this approach. So we need a set of direction vectors that can uniformly partition the degenerate Pareto front. Although we get some location vectors by vector-based effective direction detection method, the number of location vectors in the degenerate Pareto front problem is far from enough to maintain the diversity of the population, so the auxiliary reference vectors is necessary. As mentioned in our previous work CA-MOEA [6], hierarchical clustering is useful to produce reference points in irregular shaped area. However, in many-objective optimization problems, as there are so many non-dominated individuals in the objective space, we need more powerful convergence method. As shown in Figs. 3 and 4a–j are ten individuals in the objective space, LV1 and LV2 are two location vectors. If we do hierarchical clustering directly among the population, individuals with the least sum of squared errors to each other will be merged into one cluster [6], so A, B and C are merged into one cluster, D is a separate cluster, E and G are merged into a new cluster, F and H are merged into one cluster, I and J are merged into a cluster, just as shown in Fig. 3. Thus we get five cluster centers by calculating the average objective values of members in each cluster. Then we can calculate five cluster center vectors CV1, CV2, CV3, CV4 and CV5 by original point and five cluster centers. It can be seen from Fig. 3, on one hand, as cluster A, B and C, cluster F and H have cluster centers too close to each other, CV2 and CV3 almost coincided. On the other hand, although I and J are very close to each other, they are actually in two completely different convergence direction. Moreover, although D is far away from E, G and I, actually they belong to the same convergence direction. Obviously, as this clustering method cannot distinguish the convergence direction of individuals, it cannot appropriately partition the convergence directions. Besides, we have also tried some other clustering methods, including angle-based clustering and so on, to find suitable partitions, but the results are not good enough.

For better partitioning, we propose a new mapping-based clustering method. Firstly, the hyper planes are calculated with the location vectors as the normal vectors. In Fig. 4, the two solid line segments represent the hyper planes with location vectors LV1 and LV2 as the normal vectors, separately. Secondly, we do mapping for each individual which are associated with location vectors. To do this, we use the origin and each individual to make a straight line, and then calculate the intersection of the straight line and the hyperplane.

Fig. 3 Cluster center vectors produced by clustering directly

Fig. 4 Cluster center vectors produced by clustering the mapped candidates

This intersection is the mapping point on the hyperplane of an individual. Thirdly, we do hierarchical clustering on the hyper planes using the Ward's linkage criterion measured by cosine distance. Fourthly, cluster centers are calculated in every cluster, and cluster center vectors are calculate by these cluster centers and the origin. By this way, individuals in the near convergence direction will be clustered into a cluster, just like the cluster consist of A and F, and the cluster constructed by D, E, G and H. Finally, we get more uniformly distributed reference vectors CV1, CV2, CV3, CV4 and CV5. These cluster center vectors cooperate with location vectors to guiding the convergence direction of population while maintaining diversity.

Proposed algorithm LC-MaOEA

In this section, we present the details of LC-MaOEA. Similar to MOEA/D [14], each individual of LC-MaOEA corresponds to a reference vector. Then, the neighborhood of each reference vector is calculated, and offspring is generated by the individuals with the neighborhood serial numbers. The PBI value is used to evaluate the individuals and do the environmental selection. A detection method is used in LC-MaOEA to predict the effective convergence directions of the Pareto optimal individuals. Thus, the environmental selection in LC-MaOEA differs from that in MOEA/D which uses fixed reference vectors generated by Das and Dennis's systematic approach. In the following, we first present the framework of LC-MaOEA, followed by a description of the main components of the algorithm.

LC-MaOEA

Figure 5 presents the algorithm flowchart of LC-MaOEA. At first, an initial parent population of size N is randomly generated according to the problem to be solved. Then, N uniformly distributed vectors V are generated by Das and Dennis's systematic approach [23]. Meanwhile, h is initialized as 0 that indicates the Pareto area has not been detected. After that, the closest b vectors of each vector are calculated as its neighbor vectors and the ordinal numbers of the neighbor vectors are stored in B . To improve the utilization of the vectors, the objective values of each individual are scaled as follows:

$$P_{i,n}^j = P_i^j - z_{\min}^j, \quad (1)$$

where, P_i^j represents the j th objective of individual P_i , $z_{\min}^j$ represents the minimum value in the j th objective of all the individuals. Then, to locate the Pareto front, every scaled individual is assigned to a vector with the smallest angle to itself, and the vector has at least one individual assigned to it is regarded as effective vector in the current iteration. When the effective vectors remain unchanged in several iterations, it is said that the effective vectors are stable. At this time, h will be set as 1 to avoid duplicate detection.

The stable effective vectors will be regarded as location vectors LV, which is used to locate the approximate position of Pareto front. Then the axis vectors $[0, \dots, 0, 1]$, $[0, \dots, 0, 1, 0], \dots, [1, 0, \dots, 0]$ are used to search boundary individuals. Since the Pareto front is degenerate, the number of LV is so limited that the population diversity cannot be guaranteed. So the cluster center vectors CV are designed to fix this issue.

The way to produce cluster center vectors CV is explained in Algorithm1. In addition, if the number of direction vectors

Fig. 5 Cluster center vectors produced by clustering the mapped candidates

is less than population size N , we generate some random vectors within the range of effective directions to complement the number of reference vectors. For generating offspring, each individual P_i will randomly choose one of its neighbor individuals from $B(i)$, and generate a new individual Q_i by simulated binary crossover (SBX) [24] and polynomial mutation [25] with a crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/D$, where, D is the number of decision variables. Here, we use the sequence number of the vector, not the sequence number of the individual itself. Since at first the vectors and individuals do not correspond one to one, this mating pattern is equivalent to the random mating pattern, which is conducive to exploring more areas of the objective space. Then, as the iteration progresses, each vector will find the individual with smaller PBI [14] values, which make it becomes a neighborhood-based enhanced mating model. In our opinion, this method is more suitable for handling many-objective problems with degenerate Pareto fronts. Because it not only has certain exploratory ability, but also can concentrate to produce more individuals in effective directions. In environmental selection, the new produced individual will be compared with each neighbor parent and the one with smaller PBI value will be selected. This main loop repeats until a termination condition is met and the population produced by the last generation will be the final population.

Cluster center vector generation

The cluster center vectors are designed to better partition the convergence direction so as to maintain the diversity of the population. As explained in Algorithm 1, for each location vector LV_i , the hyperplane H_i with LV_i as normal vector is calculated. Thus we obtain several piecewise hyperplanes. After that, all the individuals P_{LV_i} assigned to LV_i are mapped to the hyperplane H_i , and be recorded as P_{map_i} . All the non-duplicate mapped individuals P_{map_i} are merged to K clusters by hierarchical cluster method. Ward's linkage criterion measured by cosine distance is used to do the hierarchical clustering. The parameter K is no more than the number of non-duplicate mapped individuals P_{map_i} . Afterwards, the clustering center of each cluster is obtained by the average value of each objective of all individuals in the cluster. Detailed hierarchical clustering method and clustering center calculation method can refer to CA-MOEA [6]. Finally, the cluster vectors CV are calculated by the origin and these cluster center points. Since this clustering method is able to evenly divide the mapped individuals into any number of groups, the clustering center vectors are able to evenly partition the convergence direction.

Algorithm 1: Cluster Center Vector (LV, M, P_{LV_i})

Input: Let LV is the Location vectors, M is the objective number, $P_{LV_i} (i = 1, \dots, |LV|)$ represents all the individuals assigned to $LV_i (i = 1, \dots, |LV|)$

Output: Cluster vector set CV

1: % Map individuals to the piecewise hyperplanes

2: **for** $i \leftarrow 1$ to $|LV|$ **do**

3:   **for** $j \leftarrow 1$ to M **do**

4:      $P_{map_i}^j = P_{LV_i}^j \times \sum_{j=1}^M (LV_i^j)^2 / \sum_{j=1}^M (P_{LV_i}^j \times LV_i^j)$

5:   **end for**

6: **end for**

7: $K \leftarrow \min(|P_{LV}| - |LV|, |P_{map}|)$

8: % Hierarchical clustering measured by cosine distance

9: % using the Ward's linkage criterion

10: $C_1, C_2, \dots, C_K \leftarrow$ Hierarchical clustering (P_{map}, K)

11: % Calculate cluster centers e_r

12: **for** $r \leftarrow 1$ to K **do**

13:    $p_r \leftarrow$ Find all individuals in cluster

14:   **for** $j \leftarrow 1$ to M **do**

15:      $e_r = \sum_{j=1}^M p_r^j / |p_r|$

16:   **end for**

17: **end for**

18: $CV \leftarrow$ Constructing vectors by $e_r (r = 1, \dots, K)$ and Origin

---

Computational complexity of LC-MaOEA

The worst-case complexity of the major operations in one generation of the proposed LC-MaOEA is analyzed as follows. First, the objective normalization requires $O(MN)$. Next, assigning N individuals to N vectors takes $O(N)$. For calculating cluster center vectors, mapping takes $O(N)$, calculating cluster centers takes $O(N^2)$. Then, generating offspring requires $O(N)$. For environmental selection, the comparison of PBI value takes $O(bN)$, b is the size of neighbors which is smaller than N .

To summarize, the computational complexity of LC-MaOEA in the worst cases is $O(N^2)$, which is comparable to MOEA/D-AM2M, RVEA*, A-NSGA-III and CA-MOEA.

Empirical results

In this section, the performance of LC-MaOEA is empirically examined on widely used scalable benchmark problems DTLZ5, DTLZ6 which have degenerate Pareto fronts

when the objective number is more than or equal to three. Meanwhile, LC-MaOEA is compared with four state-of-the-art algorithms designed for irregular Pareto front, namely, MOEA/D-AM2M, RVEA*, A-NSGA-III and CA-MOEA. All experiments are run on PlatEMO [26].

Performance indicators

The inverted generational distance (IGD) [27] is adopted as performance indicator in this work, which is calculated as follows:

$$IGD(\mathbf{A}, \mathbf{Z}_{\text{eff}}) = \frac{1}{|\mathbf{Z}_{\text{eff}}|} \sum_{i=1}^{|\mathbf{Z}_{\text{eff}}|} \min_{j=1}^{|\mathbf{A}|} d(\mathbf{z}_i, \mathbf{a}_j), \quad (2)$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots]$ is the non-dominated individuals obtained by an MOEA in the objective space, $\mathbf{Z}_{\text{eff}} = [\mathbf{z}_1, \mathbf{z}_2, \dots]$ is a set of solutions sampled from the known Pareto front. $d(\mathbf{z}_i, \mathbf{a}_j) = \|\mathbf{z}_i - \mathbf{a}_j\|_2$ is the Euclidean distance from $\mathbf{z}_i$ to $\mathbf{a}_j$. The IGD metric is able to measure both diversity and convergence of $\mathbf{A}$ if $|\mathbf{Z}_{\text{eff}}|$ is large enough. $|\mathbf{Z}_{\text{eff}}|$ here is set to 10,000 for 2 and 3 objective problems. Note that a smaller IGD value indicates a better performance. As the degenerate front distributes in a very narrow area of the objective space, and the widely used HV indicator [28] is calculated by hypervolume, so there will be a mistake that poorly converged populations have better HV values. Based on above consideration, we don't use HV indicator to measure the results.

Experimental settings

The algorithms are implemented in MATLAB R2014a and run on a PC with Intel Core i7-6500U, CPU 2.50 GHz, RAM of 12.00 GB. The population size is set according to $N = C_{H+m-1}^{m-1}$ [18], where, H specifies the granularity or resolution of weight vectors by Das and Dennis's systematic approach, m represents the number of objectives. Considering the number of objectives and the parameter settings in the existing literature, we set the population size to 126 for 5-objective problems, 275 for 10-objective problems, 240 for 15-objective problems, 230 for 20-objective problems, 465 for 30-objective problems and 820 for 40-objective problems. We do not test the problems with more than 40 objectives, but we think the algorithm can solve these problems with more than 40 objectives. The maximum iteration numbers are set to 250 for DTLZ5 and DTLZ6. These parameter settings are the same as recommended in [29] and [30]. The simulated binary crossover (SBX) [24] and polynomial mutation [25] are adopted in LC-MaOEA with a crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/D$, where, D represents the number of decision variables. Meanwhile, the

Table 1 the statistical results (mean and standard deviation) of IGD values on 5–40 objectives DTLZ5 and DTLZ6 problems calculated by MOEA/D-AM2M, RVEA*, A-NSGA-III, CA-MOEA and LC-MaOEA

Pro.	Obj.	MOEA/D-AM2M	RVEA*	A-NSGA-III	CA-MOEA	LC-MaOEA
		Mean	Mean	Mean	Mean	Mean
		Std.	Std.	Std.	Std.	Std.
DTLZ5	5	2.16E-02(2)+	5.72E-02(3)+	1.52E-02	2.86E-01(5)+	2.00E-02(1)
	10	1.71E-02(2)+	1.06E-01(3)+	2.34E-02	7.17E-01(5)+	5.15E-03(1)
	15	3.82E-02(2)+	1.37E-01(3)+	3.25E-02	7.81E-01(5)+	7.27E-03(1)
	20	5.85E-02(2)+	1.51E-01(3)+	2.39E-02	8.14E-01(5)+	4.85E-03(1)
	30	4.99E-02(2)+	1.80E-01(3)+	2.42E-02	8.84E-01(5)+	3.97E-03(1)
DTLZ6	40	4.93E-02(2)+	1.79E-01(3)+	3.07E-02	8.13E-01(5)+	2.39E-03(1)
	5	1.25E-01(3)+	1.22E-01(2)+	3.81E-02	1.55E+00(5)+	2.63E-02(1)
	10	1.78E-02(2)+	1.89E-01(3)+	3.55E-02	6.96E+00(5)+	4.47E-03(1)
	15	5.77E-02(2)+	2.13E-01(3)+	4.30E-02	8.25E+00(5)+	4.96E-03(1)
	20	1.81E-01(2)+	2.51E-01(3)+	6.37E-02	8.19E+00(5)+	2.92E-03(1)
	30	1.42E-01(2)+	2.28E-01(3)+	3.93E-02	7.41E+00(5)+	5.66E-02(1)
	40	7.22E-02(2)+	2.27E-01(3)+	5.45E-02	7.61E+00(5)+	5.53E-03(1)

(1)(2)(3)(4)(5)(6)(7): The rank of IGD value, the smaller the rank value, the better the performance, +: LC-MaOEA shows significantly better performance in the comparison, -: LC-MaOEA shows significantly worse performance in the comparison, ≈: There is no significant difference between the compared results

distribution index of crossover is set to and the distribution index of mutation, as recommended in [30].

IGD results on the test suites

Table 1 present the IGD results of MOEA/D-AM2M, RVEA*, A-NSGA-III, CA-MOEA and LC-MaOEA on test problems with 5–40 objectives. The IGD values are averaged over 20 independent runs. The mean values for each instance are presented and the standard deviation are listed. The results obtained by LC-MaOEA are shaded if they are the best among the five compared algorithms. For more rigorous comparisons, the Wilcoxon rank sum test is used to compare the result of the algorithms at a significance level of 0.05. Symbol ‘+’ represents that LC-MaOEA significantly outperforms the compared algorithm, while symbol ‘–’ indicates the compared algorithm performs significantly better than LC-MaOEA. Meanwhile, symbol ‘ $\approx$ ’ states there is no significant difference between these two algorithms.

It can be found from Table 1 that LC-MaOEA achieves the best performance on all the twelve test problems of DTLZ5 and DTLZ6. It is noteworthy that our algorithm improves the results of these two problems by more than one order of magnitude. Moreover, MOEA/D-AM2M reaches the second best results on eleven out of twelve test problems of DTLZ5 and DTLZ6, which shows its ability on handling this kind of problems.

In addition, the performance of RVEA* and A-NSGA-III on these test problems are mediocre. On the one hand, RVEA* achieves the second best result on 5-objective DTLZ6 problem and the third place on rest five test problems of DTLZ6. It also reaches the third place on all the DTLZ5 test problems. On the other hand, A-NSGA-III get the fourth place on all the test problems. These results may due to the fact that these two algorithms are constantly adjusting the convergence direction according to the current population, so that they are easy to be deceived, and their exploration ability and convergence pressure are insufficient.

Figure 6 shows the results of 10-objective DTLZ6 problems by Pareto optimal solutions, MOEA/D-AM2M, RVEA*, A-NSGA-III, CA-MOEA and LC-MaOEA. Abscissa of the figures is the objective order number and longitudinal coordinates are the objective values of each individual on each objective. Each large graph has the same coordinate range. It should be noted, however, that since the range of the longitudinal coordinates of Pareto front and LC-MaOEA are relatively small, we have enlarged the effective area with a small graph with the range of the longitudinal coordinates from 0 to 1. It can be seen from Fig. 6 that our proposed LC-MaOEA can well approximate the Pareto front of 10-objective DTLZ6 problem, while none of the other algorithms can converge near the Pareto front. But among them, we can see that MOEA/D-AM2M find the most Pareto optimal individuals, although there are still many dominated individuals. Apart from LC-MaOEA, the results of RVEA*

Fig. 6 a Pareto optimal solutions of 10-objective DTLZ6 and the optimization results of 10-objective DTLZ6 obtained by b MOEA/D-AM2M, c RVEA*, d A-NSGA-III, e CA-MOEA, f LC-MaOEA

Fig. 7 Convergence rate on **a** 10-objective DTLZ5, **b** 15-objective DTLZ6

converge in the range closest to Pareto front, nevertheless, it can find very few Pareto optimal individuals so that the Pareto front area is almost blank. Moreover, the results of A-NSGA-III are even worse which can be attributed to the ineffective update mechanism of the weight vectors and the slow convergence rate due to frequent changes in the reference direction.

The fast non-dominated sorting based CA-MOEA performs the worst among the five compared algorithms in that the final population are far away from the PF. The reason why these comparison algorithms cannot simulate Pareto front of DTLZ6 well is that DTLZ6 is a highly misleading nonlinear conflict problems [22], so these algorithms lack strong recognition and convergence ability to achieve satisfactory results. Overall, it can be seen that LC-MaOEA shows significant better capability in approximating the Pareto fronts of DTLZ5 and DTLZ6 problems compared to the other state-of-the-art algorithms.

Comparison on convergence

We further compare the performance of MOEA/D-AM2M, RVEA*, A-NSGA-III, CA-MOEA and LC-MaOEA by examining their convergence speed.

Figure 7 demonstrates the curves of IGD value over iterations on (a) 10-objective DTLZ5, (c) 15-objective DTLZ6, respectively. The red circles denote the change in IGD values of LC-MaOEA, and the black points, crosses, stars, triangles denote that of MOEA/D-AM2M, RVEA*, A-NSGA-III, and CA-MOEA, respectively. As shown in Fig. 7, LC-MaOEA converges faster in the earlier generations on all of these test problems with degenerate Pareto fronts, and then stayed at a stable level, which owe to its stable convergence direction.

Optimization of the polyester filament melt-transportation process

Polyester filament has the characteristics of high strength, good heat resistance, good elasticity, wrinkle resistance, wear

Fig. 8 Schematic diagram of polyester filament melt-transportation process

resistance and good insulation. It has been widely used in various fields of national economy. Melt-transportation is an important process in polyester filament production. The schematic diagram of melt transportation process is demonstrated in Fig. 8.

According to [31], five performance indexes are selected as the optimization objectives, namely, the intrinsic viscosity f_1 , temperature f_2 , pressure of the melt in spinneret inlet pipe f_3 , the residence time of the melt during the whole process f_4 and the pressure of the melt in metering pump inlet pipe f_5 . Six decision variables affect the objective function, namely, initial intrinsic viscosity of the melt IV_0 , initial temperature T_0 , initial pressure P_0 , initial flow rate G , temperature of the heat-transfer medium pump T_{sb} , and the small heat-transfer pump used in the melt-transportation process T_{ss} . The objective function formulas are given in (3).

$$\begin{cases} \min f_1(\mathbf{x}) = |y_{\Delta IV} - \Delta IV| \\ \min f_2(\mathbf{x}) = |y_{\Delta T} - \Delta T| \\ \min f_3(\mathbf{x}) = |y_{\Delta P} - \Delta P| \\ \min f_4(\mathbf{x}) = |y_{P_b} - P_b| \\ \min f_5(\mathbf{x}) = y_{\text{time}} \end{cases} \quad (3)$$

where, $\mathbf{x} = (IV_0, T_0, P_0, G, T_{sb}, T_{ss})$, $y_{\Delta IV}$ represents mechanism model output for reducing melt intrinsic viscosity, $y_{\Delta T}$ is the increase of the melt temperature during the whole process, $y_{\Delta P}$ represents mechanism model output for reducing melt pressure, y_{P_b} represents the mechanism model output of the pressure of the melt in spinneret inlet pipe, y_{time} represents the output of the melt residence time of the whole process. Meanwhile, ΔIV , ΔT , ΔP , P_b represent the expectation value of $y_{\Delta IV}$, $y_{\Delta T}$, $y_{\Delta P}$, y_{P_b} , respectively. This is a 5-objective optimization problem. We compare the performance of MOEA/D-AM2M, RVEA*, A-NSGA-III, CA-MOEA and LC-MaOEA. The parameter set are the same as suggested in [31]. Figure 9 shows the results of melt-transportation process obtained by reference points given by [31], MOEA/D-AM2M, RVEA*, A-NSGA-III, CA-MOEA

Fig. 9 **a** Reference points of melt-transportation process and the optimization results obtained by **b** MOEA/D-AM2M, **c** RVEA*, **d** A-NSGA-III, **e** CA-MOEA, **f** LC-MaOEA

and LC-MaOEA. Since the range of vertical coordinates of each result is too different, we have not unified them. So as shown in Fig. 9, the maximum objective value of MOEA/D-AM2M is 5, RVEA* is 7, A-NSGA-III is 8, CA-MOEA is 25, the reference points and LC-MaOEA is 0.8. We can see that LC-MaOEA can best simulate the reference results. What's more, according to [31], the smaller the results, the better. As the results of LC-MaOEA concentrate in the smallest range, it quite meet the requirement.

In conclusion, LC-MaOEA performs well on real-world application with irregular Pareto front which are difficult to converge.

Conclusion

In this paper, a many-objective evolutionary algorithm based on Pareto front location and piecewise mapping clustering (LC-MaOEA) is proposed for solving a class of MaOPs with degenerate Pareto fronts. The main idea of the proposed algorithm is to guide the population to converge along more efficient and promising directions. To achieve this, a Pareto front detection method is proposed to find the

reliable convergence direction, and a mapping-based hierarchical clustering algorithm is developed to better partition the convergence direction. Meanwhile, a neighbor-based mating selection method is used to produce more individuals in the effective convergence directions. The proposed LC-MaOEA is compared with four state-of-the-art algorithms designed for degenerate problems, and tested on DTLZ5 and DTLZ6 problems with 5–40 objectives. In addition, LC-MaOEA is applied to optimize the parameters in polyester filament melt-transportation process. The results on both benchmark problems as well as the application demonstrate that LC-MaOEA is able to obtain satisfactory solutions on MaOPs with degenerate Pareto fronts.

Despite its attractive performance demonstrated on both test problems and a real-world application, LC-MaOEA has much room for improvement in that it is competitive mainly for a particular class of MaOPs with degenerated problems. Our future work includes extending LC-MaOEA to solve problems with various shapes of Pareto fronts.

Acknowledgements Yicun Hua would like to thank Houyue Xu for providing the Matlab code for optimization of polyester filament melt-transportation process. This work was supported in part by the National Key Research and Development Plan from Ministry of Science and Technology (2016YFB0302701), Natural Science Foundation of Shanghai (19ZR1402300), the Fundamental Research Funds for the Central Universities (No. 2232017D-13, No. 2232020D-48), Agricultural Project of the Shanghai Committee of Science and Technology (No. 16391902800), and National Nature Science Foundation of China (No. 61903078).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Li K, Wang R, Zhang T, Ishibuchi H (2018) Evolutionary many-objective optimization: a comparative study of the state-of-the-art. *IEEE Access* 6:26194–26214
- Wang H, Jiao L, Yao X (2015) Two_Arch2: an improved two-archive algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(4):524–541
- Yu G, Chai T, Luo X (2011) Multiobjective production planning optimization using hybrid evolutionary algorithms for mineral processing. *IEEE Trans Evol Comput* 15(4):487–514
- Chand S, Wagner M (2015) Evolutionary many-objective optimization: a quick-start guide. *Surv Oper Res Manag Sci* 20(2):35–42

5. Ishibuchi H, Setoguchi Y, Masuda H, Nojima Y (2017) Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Trans Evol Comput* 21(2):169–190
6. Hua Y, Jin Y, Hao K (2019) A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular pareto fronts. *IEEE Trans Cybern* 49(7):2758–2770
7. Lin Q, Liu S, Wong K, Gong M, Coello CAC, Chen J, Zhang J (2019) Clustering-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans Evol Comput* 23(3):391–405
8. Pan L, He C, Tian Y, Su Y, Zhang X (2017) A region division based diversity maintaining approach for many-objective optimization. *Integrat Comput Aided Eng* 24(3):279–296
9. He C, Tian Y, Jin Y, Zhang X, Pan L (2017) A radial space division based evolutionary algorithm for many-objective optimization. *Appl Soft Comput* 61:603–621
10. Pan L, Li L, He C, Tan KC (2019) A subregion division-based evolutionary algorithm with effective mating selection for many-objective optimization. *IEEE Trans Cybern* (Early Access). <https://doi.org/10.1109/TCYB.2019.2906679>
11. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
12. Liu Q, Jin Y, Heiderich M, Rodemann T (2019) Adaptation of reference vectors for evolutionary many-objective optimization of problems with irregular Pareto fronts. *IEEE congress on evolutionary computation*, Wellington, New Zealand
13. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: handling constraints and extending to an adaptive approach. *IEEE Trans Evol Comput* 18(4):602–622
14. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
15. Qi Y, Ma X, Liu F, Jiao L, Sun J, Wu J (2014) MOEA/D with adaptive weight adjustment. *Evol Comput* 22(2):231–264
16. Gu F, Cheung YM (2018) Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm. *IEEE Trans Evol Comput* 22(2):211–225
17. Cai X, Mei X, Fan Z (2018) A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors. *IEEE Trans Cybern* 48(8):2335–2348
18. Cai X, Yang Z, Fan Z, Zhang Q (2017) Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization. *IEEE Trans Cybern* 47(9):2824–2837
19. Liu H-L, Chen L, Zhang Q, Deb K (2018) Adaptively allocating search effort in challenging many-objective optimization problems. *IEEE Trans Evol Comput* 22(3):433–448
20. Ge H, Zhao M, Sun L, Wang Z, Tan G, Zhang Q, Chen CLP (2019) A many-objective evolutionary algorithm with two interacting processes: cascade clustering and reference point incremental learning. *IEEE Trans Evol Comput* 23(4):572–586
21. Ishibuchi H, Masuda H, Nojima Y (2016) Pareto fronts of many-objective degenerate test problems. *IEEE Trans Evol Comput* 20(5):807–813
22. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R (eds) *Evolutionary multiobjective optimization: theoretical advances and applications*. Springer, Berlin, pp 105–145
23. Das I, Dennis J (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J Optim* 8(3):631–657
24. Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9(3):115–148
25. Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inform* 26(4):30–45
26. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87
27. Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput* 12(1):41–63
28. While L, Hingston P, Barone L, Huband S (2006) A faster algorithm for calculating hypervolume. *IEEE Trans Evol Comput* 10(1):29–38
29. Ishiuchi H, Masuda H, Nojima Y (2016) Pareto fronts of many-objective degenerate test problems. *IEEE Trans Evol Comput* 20(5):807–813
30. Zhang X, Tian Y, Jin Y (2015) A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(6):761–776
31. Xu H, Hao K, Chen L, Cai X, Ren L, Ding Y (2018) Craft parameters optimization of melt-transportation in polyester fiber production based on improved RVEA. 2018 IEEE 7th data driven control and learning systems conference (DDCLS), pp 356–361

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.