Improving 1by1EA to Handle Various Shapes of Pareto Fronts

Yiping Liu[1], Hisao Ishibuchi[2], Yusuke Nojima[1(✉)], Naoki Masuyama[1], and Ke Shang[2]

[1] Department of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan
yiping0liu@gmail.com, {nojima,masuyama}@cs.osakafu-u.ac.jp
[2] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, Guangdong, China
hisao@sustc.edu.cn, kshang@foxmail.com

Abstract. 1by1EA is a competitive method among existing many-objective evolutionary algorithms. However, we find that it may fail to find boundary solutions depending on the Pareto front shape. In this study, we present an improved version of 1by1EA, named 1by1EA-II, to enhance the flexibility in handling various shapes of Pareto fronts. In 1by1EA-II, the Chebyshev distances from a solution to the nadir and ideal points are alternately employed as two convergence indicators. Using the first convergence indicator, boundary solutions are preferred for a wide spread in the objective space. With the other convergence indicator, non-boundary solutions are preferred to promote diversity. We empirically compare the proposed 1by1EA-II with its original version as well as four other state-of-the-art algorithms on DTLZ and Minus-DTLZ test problems. The results show that 1by1EA-II is the most flexible algorithm.

Keywords: Many-objective evolutionary computation
Pareto front shape · Convergence · Diversity

1 Introduction

There exist a large number of multi-objective optimization problems (MOPs) in real-world applications. The conflict of objectives implies that there is no single optimal solution to an MOP, rather a set of trade-off solutions, called the Pareto optimal solution set (PS). The image of PS in the objective space is referred to as the Pareto front (PF). Without loss of generality, an MOP can be mathematically expressed as follows:

$$\begin{aligned} &\min \boldsymbol{f}(\boldsymbol{x}) = \min(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})) \\ &\text{s.t. } \boldsymbol{x} \in S \subset \boldsymbol{R}^n \end{aligned} \tag{1}$$

A. Auger et al. (Eds.): PPSN 2018, LNCS 11101, pp. 311–322, 2018.
https://doi.org/10.1007/978-3-319-99253-2_25

where $\boldsymbol{x} = (x_1, \ldots, x_n)$ represents an n-dimensional decision vector in space S, $f_m(\boldsymbol{x}), m = 1, \ldots, M$, is the m-th objective to be minimized, and M is the number of objectives. When $M > 3$, the problem in Eq. (1) is referred to as a many-objective optimization problem (MaOP).

Multi-Objective Evolutionary Algorithms (MOEAs) are widely applied to solve MOPs, where the Pareto dominance-based ones are most popular, such as Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4] and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [17]. However, their performance generally deteriorates appreciably for MaOPs. One main reason is the low efficiency of the Pareto dominance-based selection strategy in the high-dimensional objective space.

To address this issue, various MOEAs aiming at solving MaOPs have been developed in recent years. Generally, they can be categorized into the following three categories: (1) improved Pareto dominance-based methods, e.g., SPEA2 with shift-based density estimation (SDE) [8] and NSGA-III [3]; (2) decomposition-based methods, e.g., MOEA/D [15] and Reference-Vector-guided Evolutionary Algorithm (RVEA) [2]; (3) indicator-based methods, e.g., Hypervolume Estimation algorithm (HypE). Besides, there are a number of novel methods that have not been categorized, e.g., Grid-based Evolutionary Algorithm (GrEA) [14], Knee point driven Evolutionary Algorithm (KnEA) [16], Bi-Goal Evolutionary approach (BiGE) [9], Reference Points-based Evolutionary Algorithm (RPEA) [11], and One-by-One selection-based Evolutionary Algorithm (1by1EA) [10].

These MOEAs often show encouraging performance on widely used benchmarks such as DTLZ [5] and WFG [6]. However, their performance may strongly depend on the PF shapes. For example, by simply inverting the PF shapes of DTLZ, the performance of a decomposition-based method noticeably degrades [7]. Similarly, NSGA-III and MOMBI-II which share the concept of decomposition would also have the issue. For another instance, the performance of some methods like GrEA is very sensitive to the parameter settings, and it is difficult to tune the parameters according to the PF shapes. The real-world optimization problems usually have various shapes of PFs. Therefore, developing more flexible MOEAs is a must, where improving the flexibility of existing state-of-the-art MOEAs is very promising.

In this study, we improve 1by1EA's ability in solving MaOPs with various shapes of PFs. 1by1EA is very competitive among existing many-objective optimizers. As shown by our computational experiments later in this paper, 1by1EA has high search ability on DTLZ (which is higher than other many-objective optimizers such as NSGA-III, MOEA/D, BiGE and KnEA). 1by1EA adopts not only a one-by-one selection strategy to well balance the convergence and the diversity of solutions, but also a corner solution preserving strategy for a wide spread. However, we find that 1by1EA may not perform well when the corner solutions are difficult to be located on a PF. In this study, we present an improved version of 1by1EA, named 1by1EA-II. It alternately employs two convergence indicators to search the boundary and non-boundary solutions and is more flexible than its original version.

The remainder of this paper is organized as follows. In Sect. 2, 1by1EA is first briefly introduced and the motivation of this work is elaborated. The proposed 1by1EA-II is then described in detail in Sect. 3. Section 4 presents experimental results and discussions. Section 5 concludes the paper and provides future research directions.

2 Preliminaries

In this section, we first briefly introduce 1by1EA [10] and then elaborate the motivation of this work.

2.1 A Brief Introduction to 1by1EA

The general framework of 1by1EA is similar to standard generational evolutionary algorithms, whereas its environmental selection makes it special.

Assume to solve the problem in Eq. (1) using 1by1EA. Before the environmental selection, the convergence and distribution indicators of each candidate solution are calculated. Please refer to the original study for the examples of these indicators [10]. The convergence indicator is usually a scalarizing function aggregating all objective functions, such as the sum of all objective functions or the Euclidean distance between the solution and the ideal (nadir) point. Note that we estimate the ideal (nadir) point in terms of the minimum (maximum) objective values among obtained non-dominated solutions in this study. The convergence indicator can provide an extremely large selection pressure towards the PF. The general formulation of the convergence indicator can be summarized as follows:

$$c(\boldsymbol{x}) = \mathrm{agg}(f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})). \tag{2}$$

The distribution indicator is the cosine similarity between the solution and each of the others. It can efficiently reduce the number of dominance resistant solutions.

Next, M corner solutions are selected to estimate the boundary of the PF. The mth corner solution $\boldsymbol{x}_m^{\mathrm{corner}}$ is obtained by the following method:

$$\boldsymbol{x}_m^{\mathrm{corner}} = \arg\min_{\boldsymbol{x}_i \in Q} c_m(\boldsymbol{x}_i), m = 1, \ldots, M, \tag{3}$$

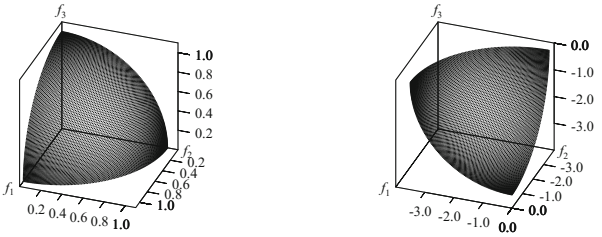
where $c_m(\boldsymbol{x}_i) = \mathrm{agg}(f_1(\boldsymbol{x}), \ldots, f_{m-1}(\boldsymbol{x}), f_{m+1}(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))$, and Q is the current population.

Finally, the one-by-one selection strategy is applied. It consists of the two important steps. In the first step, only one solution with the best value of the convergence indicator is selected, focusing on the convergence. In the second step, solutions close to the one selected in the first step are de-emphasized according to the distribution indicator, thus maintaining the diversity of the population. By repeating the above two steps, a solution set with good convergence and diversity can be obtained.

1by1EA has been demonstrated to be a competitive many-objective optimizer. However, we find that 1by1EA is not flexible enough due to the corner solution preserving strategy. The motivation of improving 1by1EA is elaborated in the next subsection.

2.2 Motivation

As reported in [7] recently, a number of newly proposed decomposition-based algorithms seem to be overspecialized for the popular benchmarks like DTLZ. In [7], the minus version of DTLZ (denoted as Minus-DTLZ) is presented, where the PF shapes are inverted from those of DTLZ. Figure 1 shows the true PFs of DTLZ2 and Minus-DTLZ2 with three objectives for intuitive understanding.



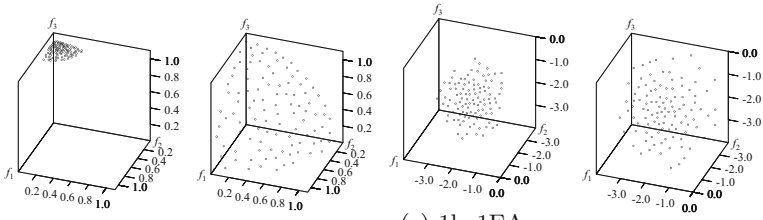
(a) DTLZ2 (b) Minus-DTLZ2

Fig. 1. The true PFs of DTLZ2 and Minus-DTLZ2 with three objectives.

The performance of decomposition-based algorithms appreciably deteriorates on Minus-DTLZ merely because the PF shapes of Minus-DTLZ are different from those of DTLZ. Some recent researches on using two reference vector sets have addressed this issue [1,13]. This inspires us to investigate the behaviors of some non-decomposition-based algorithms like 1by1EA when handling various shapes of PFs.

The corner solution preserving strategy plays an important role in 1by1EA. Figure 2 presents the solution sets obtained by 1by1EA with and without preserving corner solutions on DTLZ2 and Minus-DTLZ2 with three objectives in a typical run, where the Euclidean distance between a solution and the ideal point is chosen as the convergence indicator. Note that in this study, the inverted generational distance (IGD) [18] of the solution set obtained in the typical run is the nearest to the average IGD over 40 runs.

We can see from Fig. 2 that preserving corner solutions can lead to the solution sets widely spread in the objective space both on DTLZ2 and Minus-DTLZ2. The solution set in Fig. 2(b) approximates well to the true PF in Fig. 1(a). However, the solution set in Fig. 2(d) fails to cover some boundary regions, comparing to the true PF in Fig. 1(b). The reason is that the corner solutions to DTLZ2 (i.e., $(1,0,0),(0,1,0),(0,0,1)$) can be easily located by Eq. (3) (i.e., minimizing



(a) 1by1EA on DTLZ2 without preserving corner solutions

(b) 1by1EA on DTLZ2

(c) 1by1EA on Minus-DTLZ2 without preserving corner solutions

(d) 1by1EA on Minus-DTLZ2

Fig. 2. The solution sets obtained on DTLZ2 and Minus-DTLZ2 with three objectives.

$f_2 + f_3$, $f_3 + f_1$, and $f_1 + f_2$, respectively), whereas those to Minus-DTLZ2 (i.e., $(-3.5, 0, 0), (0, -3.5, 0), (0, 0, -3.5)$) cannot. Of course, we can use another method (i.e., by minimizing f_1, f_2, and f_3, respectively) to obtain the corner solutions of Minus-DTLZ2. However, for real-world optimization problems, we usually cannot obtain the *a priori* knowledge of corner points and apply a proper method to locate all of them. Furthermore, even if we use both of the above-mentioned methods, the shape of a PF could be too complex to locate the corner solutions.

From these observations, we notice that if the corner solution can be precisely located, 1by1EA performs perfectly; otherwise, it may miss some boundary regions on a PF. This indicates that the performance of 1by1EA also depends on the shapes of PFs.

In view of this, we present an improved version of 1by1EA, named 1by1EA-II, to enhance its flexibility in handling various shapes of PFs. In 1by1EA-II, the corner solutions are no longer needed to be preserved. The diversity of solutions are promoted by alternately employing two different convergence indicators. The details of 1by1EA-II are described in the next section.

3 1by1EA-II

The difference between 1by1EA and 1by1EA-II is that in the environmental selection of 1by1EA-II, the corner solutions are no longer preserved by Eq. (3), and two convergence indicators are alternately employed to select the solution with best convergence performance. That is, after selecting a solution according to a convergence indicator, the next solution to be selected is based on the other convergence indicator. Please refer to the original study of 1by1EA for the environmental selection procedure [10]. We describe the two convergence indicators used in 1by1EA-II in the following parts.

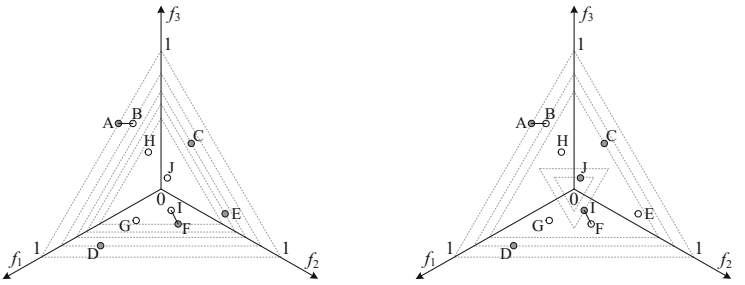
The first convergence indicator adopted in 1by1EA-II is the Chebyshev distance between a solution and the nadir point. It is formulated as follows:

$$c^{\mathrm{CdN}}(\boldsymbol{x}) = \max_{1 \le m \le M} |f_m(\boldsymbol{x}) - z_m^{\mathrm{nad}}|, \tag{4}$$

where $\boldsymbol{z}^{\mathrm{nad}} = (z_1^{\mathrm{nad}}, \ldots, z_M^{\mathrm{nad}})$ is the nadir point. Note that there is no weight in the convergence indicators, since all the objectives are equally considered in this study. The solution farthest from the nadir point is supposed to have the best convergence performance. However, since the nadir point is estimated based on the obtained non-dominated solutions, a solution dominated by the estimated nadir point may be better than others according to Eq. (4). This situation should be avoided. In addition, we want to minimize the convergence indicator. Therefore, we modify Eq. (4) into the following formulation:

$$c_1(\boldsymbol{x}) = \min_{1 \le m \le M} (f_m(\boldsymbol{x}) - z_m^{\mathrm{nad}}). \tag{5}$$

By minimizing c_1 in 1by1EA-II, the boundary solutions (including the corner solutions) are preferred (no matter the Pareto front is convex or concave). Figure 3(a) presents an illustration of the solution selection procedure only using c_1.



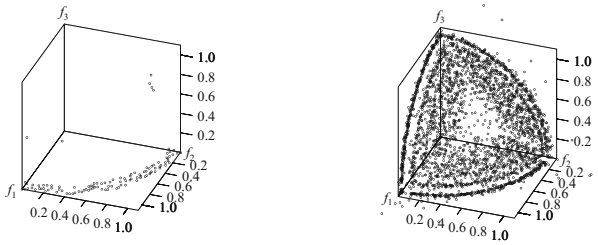
(a) Only using c_1 (b) Alternately using c_1 and c_2

Fig. 3. Solution selection procedure in 1by1EA-II.

In Fig. 3, assume (0, 0, 0) and (1, 1, 1) are the ideal and nadir points, respectively. Dots A-J are candidate solutions, and we want to select five of them into the next generation. The dashed triangles are the intersections of the contour lines of c_1 and the hyperplan defined by $f_1 + f_2 + f_3 = 1$ (note that the solutions are not necessarily on the hyperplan). Solutions connected with each other are in the other's niche according to the diversity indicator. As can be seen from Fig. 3(a), the boundary solution A has the minimum value of c_1 and all the other solutions are within the corresponding dashed triangle of A. Therefore, A is selected first. Then B is de-emphasized since it is too close to A. Next, C, D, E, and F are selected one by one due to their minimum c_1 values among the

rest. From this illustration, we can observe that boundary solutions are always preferred by minimizing c_1 no matter what is the shape of a PF. Consequently, preserving corner solutions is unnecessary in 1by1EA-II.

However, employing c_1 as the only convergence indicator may results in two issues. The first issue is that it may lead the population into a partial region of the PF, since the estimated nadir point is usually quite different from the true one at the early stage of evolution. Minimizing c_1 with an incorrect nadir point will result in solutions located in partial regions. Conversely, the nadir point in the next generation could be estimated more incorrectly by these solutions. Figure 4(a) shows the obtained solution set on DTLZ2 with three objectives in a typical run when c_1 is employed as the only convergence indicator. We can see that most solutions locate in a small region. The other issue is that even if we use the true nadir point, the solutions are more likely to locate in the boundary region. Figure 4(b) shows all the solution sets obtained in 40 runs where the true nadir point is used in c_1. We can observe that the solutions in the boundary region are denser than those in the central region.



(a) In a typical run using the estimated nadir point

(b) In 40 runs using the true nadir point

Fig. 4. The solution set obtained on DTLZ2 with three objectives when c_1 is employed as the only convergence indicator.

In view of this, we employ the Chebyshev distance between a solution and the ideal point as the other convergence indicator, which is formulated as follows:

$$c_2(\boldsymbol{x}) = \max_{1 \le m \le M} |f_m(\boldsymbol{x}) - z_m^*|, \tag{6}$$

where $\boldsymbol{z}^* = (z_1^*, \ldots, z_M^*)$ is the ideal point. In contrast to minimizing c_1, non-boundary solutions are preferred when minimizing c_2. Alternately employing c_1 and c_2 is helpful to promote diversity, since both non-boundary and boundary solutions have a chance to be selected. Let us see Fig. 3(b) as an example, where the dashed inverted triangles are the intersections of the contour lines of c_2 and the hyperplan defined by $f_1 + f_2 + f_3 = 1$. The solution J has the minimum value of c_2 and all the other solutions are outside the corresponding dashed inverted triangle of J. In this case, A, J, C, I, and D are selected one by one, and B and F

are de-emphasized after selecting A and I, respectively. There are more solutions in the central region in Fig. 3(b) than that in Fig. 3(a). In addition, solutions selected by alternately using c_1 and c_2 have a much lower chance to converge into a partial region, and the nadir point could be estimated more precisely.

By the cooperation among the above-mentioned two convergence indicators and the distribution indicator, the one-by-one selection strategy in 1by1EA-II is expected to locate the boundary solutions on a PF and maintain a good diversity within the boundary.

4 Experiments and Discussions

In this section, we empirically evaluate and discuss the performance of 1by1EA-II by comparing it with 1by1EA, NSGA-III, MOEA/D, BiGE and KnEA. DTLZ1 to 4 and Minus-DTLZ1 to 4 are chosen as test problems. We consider these test problems with 3, 4, 6, and 8 objectives. The number of variables n is set to $M + 4$ for DTLZ1 and Minus-DTLZ1, and $M + 9$ for the other test problems (M is the number of objectives). NSGA-III and MOEA/D are supposed to be overspecialized for DTLZ test problems according to [7]. No study has shown that BiGE and KnEA are overspecialized so far.

For all compared algorithms, simulated binary crossover and polynomial mutation are used as the crossover and mutation operators, with both distribution indexes being set to 20. The crossover and mutation probabilities are 1.0 and $1/n$, respectively. The population size N is set to 105, 120, 132 and 156 when M is 3, 4, 6, and 8, respectively. In 1by1EA, the Euclidean distance between a solution and the ideal point is chosen as the convergence indicator. In MOEA/D, the PBI method with $\theta = 5$ is adopted. In KnEA, T is set to 0.5. Each algorithm is run for 40 times on each test problem, where the termination condition is set to 600 generations for DTLZ3 and Minus-DTLZ3, and 300 generations for the other test problems. The source codes of 1by1EA and 1by1EA-II can be downloaded from https://github.com/yiping0liu. All the other compared algorithms are implemented by PlatEMO [12].

Table 1 lists the average values of IGD over 40 runs in gray scale, where a darker tone corresponds to a larger average value of IGD. Note that in this study the reference points for IGD calculation are uniformly sampled on a true PF, and the number of reference points is around 10^4. In Table 1, "Rank^1", "Rank^{-1}", and "Rank$^{\text{all}}$" denote the average ranks of each algorithm according to the average IGD values on DTLZ, Minus-DTLZ, and all the test problems, respectively. DTLZn-m (DTLZn^{-1}-m) denotes DTLZn (Minus-DTLZn) with m objectives. "†" indicates that the result is significantly different from that of 1by1EA-II by Wilcoxon's rank sum test where the null hypothesis is rejected at a significant level of 0.05. "+", "−", and "=" indicate the number of test problems where 1by1EA-II shows significantly better, worse, and similar performance, respectively.

From Table 1, we can see that 1by1EA-II, 1by1EA, NSGA-III, and MOEA/D generally achieve satisfactory results on DTLZ, where 1by1EA obtains the best

Table 1. Average IGD obtained by different algorithms.

IGD	1by1EA-II	1by1EA	NSGA-III	MOEA/D	BiGE	KnEA
DTLZ1-3	5.449E-2	4.834E-2 †	1.940E-2 †	1.943E-2 †	3.507E-2 †	5.564E-2 †
DTLZ1-4	4.599E-2	4.639E-2	4.186E-2 †	4.168E-2 †	8.023E-2 †	1.339E-1 †
DTLZ1-6	8.035E-2	8.406E-2 †	1.070E-1 †	8.048E-2	2.005E-1 †	2.432E-1 †
DTLZ1-8	1.056E-1	1.116E-1 †	1.610E-1 †	8.979E-2 †	3.508E-1 †	9.064E-1 †
DTLZ2-3	7.544E-2	5.143E-2 †	5.032E-2 †	5.031E-2 †	7.764E-2	6.645E-2
DTLZ2-4	1.316E-1	1.184E-1 †	1.212E-1 †	1.212E-1 †	1.640E-1 †	1.420E-1 †
DTLZ2-6	2.623E-1	2.499E-1 †	2.577E-1 †	2.558E-1 †	3.140E-1 †	2.809E-1 †
DTLZ2-8	3.692E-1	3.492E-1 †	3.619E-1	3.156E-1 †	4.006E-1 †	3.837E-1 †
DTLZ3-3	8.314E-2	5.053E-2 †	5.151E-2 †	5.163E-2 †	9.779E-2 †	1.135E-1 †
DTLZ3-4	1.345E-1	1.246E-1	1.232E-1	1.560E-1 †	2.709E-1 †	2.792E-1 †
DTLZ3-6	2.702E-1	2.591E-1 †	5.501E-1 †	4.295E-1 †	1.734E+0 †	1.651E+0 †
DTLZ3-8	3.839E-1	3.604E-1 †	1.267E+0 †	5.105E-1 †	1.465E+1 †	6.748E+1 †
DTLZ4-3	9.048E-2	6.865E-2 †	9.944E-2 †	3.826E-1 †	1.084E-1 †	9.513E-2 †
DTLZ4-4	1.656E-1	1.205E-1 †	1.764E-1 †	4.217E-1 †	1.659E-1	1.705E-1 †
DTLZ4-6	2.717E-1	2.835E-1	2.854E-1 †	5.511E-1 †	3.112E-1 †	3.105E-1 †
DTLZ4-8	3.763E-1	3.522E-1 †	3.452E-1 †	6.229E-1 †	3.980E-1 †	3.753E-1
Rank^{1}	3.0	2.0	2.9	3.1	5.1	5.0
+/-/=	\	2/11/3	7/7/2	7/8/1	13/1/2	14/0/2
DTLZ1^{-1}-3	2.376E+1	4.686E+1 †	3.056E+1 †	3.850E+1 †	2.980E+1 †	4.453E+1 †
DTLZ1^{-1}-4	4.651E+1	7.525E+1 †	7.991E+1 †	6.203E+1 †	5.555E+1 †	6.800E+1 †
DTLZ1^{-1}-6	8.708E+1	1.540E+2 †	1.256E+2 †	1.661E+2 †	9.844E+1 †	9.990E+1 †
DTLZ1^{-1}-8	1.666E+2	2.526E+2 †	1.544E+2 †	2.645E+2 †	1.244E+2 †	1.199E+2 †
DTLZ2^{-1}-3	2.345E-1	3.358E-1 †	2.390E-1	2.432E-1 †	3.262E-1 †	2.368E-1
DTLZ2^{-1}-4	4.589E-1	7.004E-1 †	5.178E-1 †	5.622E-1 †	6.616E-1 †	5.704E-1 †
DTLZ2^{-1}-6	9.169E-1	1.365E+0 †	1.121E+0 †	1.241E+0 †	1.332E+0 †	1.225E+0 †
DTLZ2^{-1}-8	1.367E+0	1.886E+0 †	1.684E+0 †	2.125E+0 †	1.879E+0 †	1.570E+0
DTLZ3^{-1}-3	1.466E+2	2.615E+2 †	1.517E+2	1.532E+2 †	1.993E+2 †	1.648E+2 †
DTLZ3^{-1}-4	2.884E+2	4.496E+2 †	3.333E+2 †	3.534E+2 †	4.167E+2 †	3.976E+2 †
DTLZ3^{-1}-6	5.717E+2	8.651E+2 †	7.039E+2 †	7.769E+2 †	8.412E+2 †	7.371E+2 †
DTLZ3^{-1}-8	7.744E+2	1.173E+3 †	1.044E+3 †	1.333E+3 †	1.185E+3 †	9.401E+2 †
DTLZ4^{-1}-3	2.214E-1	5.638E-1 †	2.348E-1 †	8.504E-1 †	2.676E-1 †	2.351E-1 †
DTLZ4^{-1}-4	5.669E-1	5.781E-1	5.160E-1 †	9.226E-1 †	5.544E-1	5.477E-1 †
DTLZ4^{-1}-6	1.109E+0	1.146E+0	1.049E+0	1.848E+0 †	1.129E+0	1.013E+0 †
DTLZ4^{-1}-8	1.454E+0	1.559E+0 †	1.403E+0 †	2.423E+0 †	1.581E+0 †	1.328E+0 †
Rank^{-1}	1.8	5.3	2.6	4.8	3.9	2.7
+/-/=	\	14/0/2	10/3/3	16/0/0	13/1/2	10/4/2
Rank^{all}	2.4	3.7	2.8	3.9	4.5	3.8
+/-/=	\	16/11/5	17/10/5	23/8/1	26/2/4	24/4/4

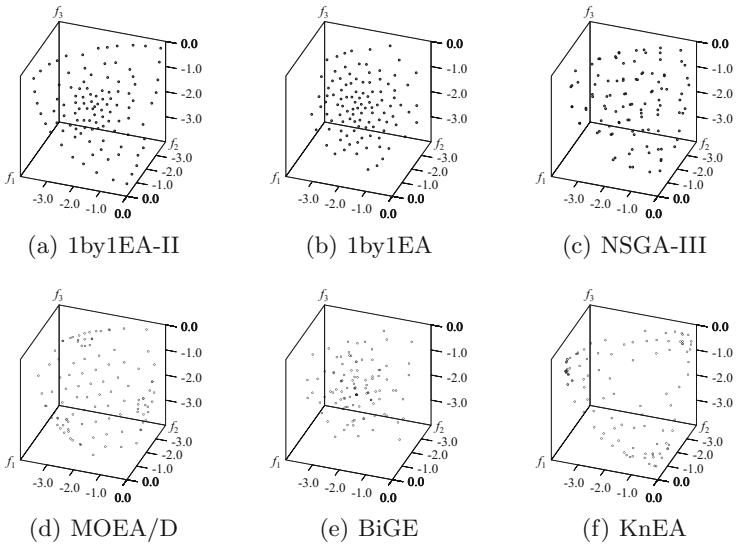
"Rank^{1}". The "Rank^{1}" of 1by1EA-II is very close to those of NSGA-III and MOEA/D, which are verified to have strength in solving DTLZ. This indicates that 1by1EA-II is very effective on solving these problems. However, 1by1EA-II does not perform as well as 1by1EA, NSGA-III, and MOEA/D on DTLZ1,

DTLZ2 and DTLZ3 with three objectives. MOEA/D performs poorly on DTLZ4, since DTLZ4 has a bias PF, which results in the failure of the PBI method in maintaining diversity in the objective space. BiGE and KnEA obtain larger values of "Rank^{1}" than the others, which suggests that they do not achieve appealing results, comparing to the algorithms that are supposed to overspecialized for DTLZ. However, they show relatively good performance on DTLZ4.

For Minus-DTLZ, 1by1EA-II outperforms the others on most test problems and obtains the best "Rank^{-1}". On the contrary, 1by1EA achieves poor IGD values on these problems and obtains the worst "Rank^{-1}". Comparing the results obtained by 1by1EA on DTLZ and Minus-DTLZ, we can notice that the performance of 1by1EA strongly depends on the PF shapes. The distribution of reference vectors (points) used in both MOEA/D and NSGA-III is inconsistent with the PF shapes of Minus-DTLZ. The performance of MOEA/D generally degrades appreciably on Minus-DTLZ, whereas the results of NSGA-III on Minus-DTLZ are still acceptable. The reason is that every reference vector (point) has to be assigned a solution in MOEA/D while it does not in NSGA-III. Moreover, in NSGA-III, multiple solutions can be clustered to one reference point, and then the reference points within the region of PF are assigned more solutions than those outside the region of PF. Consequently, the diversity of the solution set can be well maintained in NSGA-III. This observation indicates that the performance of NSGA-III is less sensitive to the PF shape than MOEA/D. The average ranks obtained by BiGE and KnEA on Minus-DTLZ are better than those on DTLZ. They achieve encouraging results on some Minus-DTLZ test problems. Theoretically, both of them are not designed to solve particular problems, however, they seem to perform better on Minus-DTLZ when comparing to the other algorithms.

As a whole, 1by1EA-II achieves the best overall performance among the compared algorithms, since it obtains the best value of "Rank^{all}". Besides, both"Rank^{1}" and "Rank^{-1}" obtained by 1by1EA-II are satisfying. Therefore, we can conclude that 1by1EA-II is most flexible among the compared algorithm on DTLZ and Minus-DTLZ test problems.

To visually demonstrate the superiority of 1by1EA-II over the other algorithms, we show the solution sets obtained by the compared algorithms in a typical run in Fig. 5. Due to space limits, only results on Minus-DTLZ2 with three objectives are presented. As can be seen from Fig. 5, 1by1EA-II can locate the boundary solutions according to the PF shape and maintain good diversity within the boundary. 1by1EA behaves as we have explained in Subsect. 2.2. For NSGA-III, there are many solutions very close to another one, since multiple solutions are clustered to one reference point. In MOEA/D, the reference vectors outside the true PF are assigned to solutions that are close to those inside the true PF, and thus the solutions obtained by MOEA/D are denser in the certain regions. The solutions obtained by BiGE fail to trace the true PF shape, and some overlap with each other. KnEA can find the boundary solutions, however, most solutions concentrate on the corner regions.



(a) 1by1EA-II (b) 1by1EA (c) NSGA-III

(d) MOEA/D (e) BiGE (f) KnEA

Fig. 5. The solution sets obtained by different algorithms on Minus-DTLZ2 with three objectives in a typical run.

5 Conclusions

In this paper, we presented an improved version of 1by1EA, named 1by1EA-II. 1by1EA-II has two distinct features. The first is that it does not preserve corner solution in the environmental selection. The other is that it alternately employs two different convergence indicators, which are the Chebyshev distances from a solution to the nadir and ideal points, respectively. By using these two convergence indicators, 1by1EA-II has an ability in achieving a well-distributed solution set according to the PF shape.

To demonstrate the effectiveness of 1by1EA-II, we tested it on DTLZ and Minus-DTLZ test problems in comparison with five state-of-the-art algorithms, namely, 1by1EA, NSGA-III, MOEA/D, BiGE, and KnEA. The experimental results demonstrated that 1by1EA-II is a competitive and flexible method among the chosen algorithms.

To further investigate and improve the flexibility of 1by1EA-II, we will apply it to optimization problems with other shapes of PFs in the future work. In addition, based on the boundary locating technique in 1by1EA-II, developing a reference vector generation method for decomposition-based algorithms is of great interest.

Acknowledgments. This work was supported by the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284).

References

1. Bhattacharjee, K.S., Singh, H.K., Ray, T., Zhang, Q.: Decomposition based evolutionary algorithm with a dual set of reference vectors. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 105–112. IEEE (2017)
2. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **20**(5), 773–791 (2016)
3. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evol. Comput. **18**(4), 577–601 (2013)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
5. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer, London (2005). https://doi.org/10.1007/1-84628-137-7_6
6. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans. Evol. Comput. **10**(5), 477–506 (2006)
7. Ishibuchi, H., Setoguchi, Y., Masuda, H., Nojima, Y.: Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. IEEE Trans. Evol. Comput. **21**(2), 169–190 (2017)
8. Li, M., Yang, S., Liu, X.: Shift-based density estimation for Pareto-based algorithms in many-objective optimization. IEEE Trans. Evol. Comput. **18**(3), 348–365 (2014)
9. Li, M., Yang, S., Liu, X.: Bi-goal evolution for many-objective optimization problems. Artif. Intell. **228**, 45–65 (2015)
10. Liu, Y., Gong, D., Sun, J., Jin, Y.: A many-objective evolutionary algorithm using a one-by-one selection strategy. IEEE Trans. Cybern. **47**(9), 2689–2702 (2017)
11. Liu, Y., Gong, D., Sun, X., Zhang, Y.: Many-objective evolutionary optimization based on reference points. Appl. Soft Comput. **50**(1), 344–355 (2017)
12. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput. Intell. Mag. **12**(4), 73–87 (2017)
13. Wang, Z., Zhang, Q., Li, H., Ishibuchi, H., Jiao, L.: On the use of two reference points in decomposition based multiobjective evolutionary algorithms. Swarm Evol. Comput. **34**, 89–102 (2017)
14. Yang, S., Li, M., Liu, X., Zheng, J.: A grid-based evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **17**(5), 721–736 (2013)
15. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
16. Zhang, X., Tian, Y., Jin, Y.: A knee point driven evolutionary algorithm for many-objective optimization. IEEE Trans. Evol. Comput. **19**(6), 761–776 (2015)
17. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm. Technical report, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) (2001)
18. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)