A Reference Points-based Evolutionary Algorithm for Many-objective Optimization

Yiping Liu
School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China
lyp.cumt@gmail.com

Dunwei Gong
School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China
dwgong@vip.163.com

Xiaoyan Sun
School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China
xysun78@126.com

Yong Zhang
School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, China
yongzh401@126.com

ABSTRACT

Many-objective optimization problems are common in real-world applications, few evolutionary optimization methods, however, are suitable for them up to date due to their difficulties. We proposed a reference points-based evolutionary algorithm (RPEA) to solve many-objective optimization problems in this study. In RPEA, a series of reference points with good performances in convergence and distribution are generated according to the current population to guide the evolution. Furthermore, superior individuals are selected based on the assessment of each individual by calculating the distances between the reference points and the individual in the objective space. The algorithm was applied to four benchmark optimization problems and compared with NSGA-II and HypE. The results experimentally demonstrate that the algorithm is strengthened in obtaining Pareto optimal set with high performances.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence – *Problem Solving, Control Methods, and Search*

Keywords

Evolutionary Optimization; many-objective optimization; reference point; distance

1. INTRODUCTION

Various optimization problems with multiple objectives commonly exist in real-world applications. Their common characteristics are that they contain more than one objective and there exist some conflicts among these objectives, indicating that there is no solution which is optimal for all objectives. They are termed as multi-objective optimization problems (MOP). Problems with more than three objectives are defined as many-objective optimization problems (MaOP). Without loss of generality, the MOP considered in this study is formulated as follows:

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2598394.2605674

$$\min \boldsymbol{f}(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_M(\boldsymbol{x})\right)$$
$$s.t. \boldsymbol{x} \in S \subset \boldsymbol{R}^n \quad (1)$$

Where $\boldsymbol{x}$ represents an n-dimensional decision variable in space S. $f_m(\boldsymbol{x}), m = 1, 2, ..., M$ is the m-th objective to be minimized, and M is the number of objectives. When M>3, this problem is a MaOP.

Classical multi-objective evolutionary algorithms (MOEA), e.g., nondominated sorting genetic algorithm II (NSGA-II) [1], have been successfully applied to a large number of MOPs. But these Pareto-based algorithms noticeably deteriorate their search ability when solving MaOPs. One major reason is that the proportion of nondominated solutions in a population rises rapidly with the number of objectives. This makes the Pareto-based selection fail to distinguish individuals. Hence, seeking for new methods to effectively solve MaOPs is of considerable necessity.

Reference points have been employed to guide the evolution in many situations. On account of the assessment of individuals by the distances between reference points and them, the selection pressure of superior individuals will not lose in many-objective optimization. Since existing reference points-based approaches usually adopt only one reference point to search solutions in objective sub-spaces of interest to the decision maker, intuitively, adopting a series of uniformly distributed reference points to obtain the whole Pareto front has the potential in solving MaOPs. Furthermore, the true Pareto front of a practical optimization problem is usually unknown, and continuously generating appropriate reference points for the problem is advantageous to achieve the solution set with perfect performance.

In view this, RPEA was proposed in this study. During the evolution, a series of reference points with good performances in convergence and distribution are adaptively generated according to the current population. Then, based on the assessment of each individual by calculating the distances between the reference points and the individual, superior individuals are selected in the environment selection process.

This study has the following contributions: (1) presenting an approach to generating reference points which are adaptive to optimization problems; (2) proposing a method of selecting superior individuals based on the distances between reference points and individuals; (3) validating the superiority of the proposed algorithm by optimizing four classical benchmark MaOPs.

The remainder of this paper is organized as follows. Section 2 reviews the related work. The proposed approach, RPEA, is presented in Section 3. Section 4 is the applications of RPEA in several benchmark MaOPs and the comparative experiments. Finally, Section 5 concludes this paper and offers suggestions on possible opportunities for future research.

2. RELATED WORK

2.1 Reference Points-based Evolutionary Optimization

Existing reference points-based approaches usually adopt only one reference point to represent the decision maker's ideal solution. Wierzbicki [2] suggested a reference point approach in which the goal is to achieve a Pareto optimal solution closest to a supplied reference point of aspiration level based on solving an achievement scalarize problem. Deb et al. [3] adopted a predator-prey approach to find a preferred set of solutions near the reference point in parallel. Mohammadi et al. [4] combined decomposition strategies with reference point approaches to search for preferred regions.

Up to date, there is little research on achieving the whole Pareto optimal solution set by employing multiple reference points. Figueira et al. [5] generated reference points by estimating the bounds of the Pareto front and searched solutions near each reference point in parallel. Deb et al. [6] adopted a family of well-distributed direction vectors to generate reference points, which makes a difference to the distribution of the solution set. Wang et al. [7] proposed a co-evolution method to simultaneously optimize solutions and reference points, but the fitness value of an individual is also calculated by the traditional Pareto dominance.

In the approach proposed in this study, reference points are suitable for different problems by continuously generating reference points based on the current population; the selection pressure in many-objective optimization is improved by calculating the distances between these reference points and individuals.

3. PROPOSED METHOD

The general ideas of RPEA are as follows. First, a series of reference points with good performances in convergence and distribution are generated according to the current population. Then, each individual in the current population and the temporary population generated by the former is assessed by calculating the distances between the reference points and the individual, and the new population is formed by selecting superior individuals.

3.1 Generation of Reference Point

In this section, the concept of reference point was first presented; a method of generating reference points with good performances in convergence and distribution was then proposed.

3.1.1 Concept of Reference Point

Loosely, the reference point is a point in the objective space that guides the evolution. There are two types of reference points, i.e., ideal point and nadir point.

For a point in the objective space, if its value(s) is (are) not inferior to that (those) of a part of the known solutions, it is termed as a local ideal point. For instance, when solving the problem represented as formula (1) with a MOEA, $\boldsymbol{r}^l=\left(r_1^l,r_2^l,...,r_M^l\right)$ is a local ideal point, when

$$r_m^l=\min_{x\in P_l} f_m\left(\boldsymbol{x}\right)+e_m, m=1,2,...,M \quad (2)$$

where P_l is a subset of the current (t-th generation) population $P(t)$. $e_m = 0 \text{or} -\varepsilon$, and ε is an arbitrarily small positive. In particular, the local ideal point

$$\boldsymbol{r}^l=\left(f_1\left(\boldsymbol{x}\right),...,f_m\left(\boldsymbol{x}\right)-\varepsilon,...,f_M\left(\boldsymbol{x}\right)\right) \quad (3)$$

must be superior to solution $\boldsymbol{x}$.

For a point in the objective space, if its value(s) is (are) not inferior to that (those) of all the known solutions, it is termed as a global ideal point. If

$$r_m^g=\min_{x\in Pg} f_m\left(\boldsymbol{x}\right)+e_m, m=1,2,...,M \quad (4)$$

where P_g is equal to $P(t)$, $\boldsymbol{r}^g=\left(r_1^g,r_2^g,...,r_M^g\right)$ is a global ideal point.

Similarly, a local nadir point and a global nadir point can also be defined. It is worth nothing that a local ideal point can become a local nadir point, and vice versa. A reference point may be a local ideal point for some solutions; however, it may be a local nadir point for other more optimal solutions, and vice versa.

3.1.2 Generation of Reference Points

If only one (local) ideal point is adopted to guide the evolution, it is difficult to obtain the whole Pareto front of a problem. But adopting a series of local ideal points instead, the whole Pareto front can be easily obtained.

While solving a problem with a MOEA, it is easy to seek for a new solution superior to the known solution in one objective. Nonetheless, it is hard to seek for a superior solution in several objectives. Therefore, a local ideal point superior in only one objective is beneficial to select solutions with good performance, and a local ideal point superior to only one solution is also helpful, compared to that superior to several solutions. To this end, formula (3) was adopted to generate reference points in this study.

The population size is supposed to be N_p. From formula (3), there exist MN_p possible reference points with better performance in convergence than the current population. However, they may not have good performance in distribution. On account of this, a method with low computation complexity for generating well diversified reference points was proposed in this section. The detailed procedure is described in algorithm1.

Algorithm1. Generation of reference points

1: $R_\Sigma\ (t)=\varnothing$

2: **for** $m=1,2,...,M$ **do**

3: Sort the individuals in $P(t)$ based on the crowding distances [1] in the m-th dimensional objective space, and select $\lceil\alpha N_p\rceil\left(\frac{N_r}{MN_p}\le\alpha\le 1\right)$ well-distributed individuals. Let $e_m=-\varepsilon$, according to formula (3), generate reference points based on the selected individuals, and form set $R_m(t)$;

4: $R_\Sigma\ (t)=R_\Sigma\ (t)\cup R_m(t)$

5: **end for**

6: Select N_r reference points from $R_\Sigma\ (t)$ based on the crowding distances in the M-dimensional objective space, and form set $R_s(t)$

When $\alpha=\frac{N_r}{MN_p}$, the aforementioned method has the lowest computation complexity, represented as $\mathrm{O}\left(MN_p\log N_p\right)$. In such scenarios, the reference points are well-distributed, and the computation complexity is substantially reduced.

3.2 Selection of Individuals

In this section, the method of selecting superior individuals was given based on the assessment of each individual by calculating the distances between the reference points and it.

In [3], a superior individual is regarded as the one close to the reference point. It is appropriate when the reference point is a local ideal point. However, for other more optimal individuals, the reference point could be a nadir point, and the closer these individuals are to the reference point, the worse performance they have. Therefore, when a reference point is employed to select individuals, it should be judged whether it is a local nadir point or not. A new method of assessing individuals was proposed in this section, which is described as follows.

The temporary population generated by the current population $P(t)$ is denoted as $Q(t)$, and the population constituted by $P(t)$ and $Q(t)$ is denoted as $P'(t)$. The p-th individual in $P'(t)$ is denoted as $\boldsymbol{x}_p, p=1,2,...,|P'|$, where $|P'|$ represents the number of individuals in $P'(t)$. The q-th reference point in $R_s(t)$ is denoted as $\boldsymbol{r}_q, q=1,2,...,N_r$. Then the distance between $\boldsymbol{x}_p$ and $\boldsymbol{r}_q$ can be formulated as

$$d(\boldsymbol{x}_p,\boldsymbol{r}_q)=\theta\sqrt{\sum_{m=1}^{M}\left(r_{qm}-f_m(\boldsymbol{x}_p)\right)^2} \tag{5}$$

where

$$\theta=\begin{cases}-1,\forall m=1,2,...,M, f_m(\boldsymbol{x}_p)\le r_{qm}\\ 1, \quad \text{other conditions}\end{cases} \tag{6}$$

Formulas (5) and (6) tell that, if $\boldsymbol{r}_q$ is a local nadir point with respect to $\boldsymbol{x}_p$, $d(\boldsymbol{x}_p,\boldsymbol{r}_q)$ represents the opposite Euclidean distance between them; otherwise, $d(\boldsymbol{x}_p,\boldsymbol{r}_q)$ represents the Euclidean distance between them. For $\boldsymbol{r}_q$, the individual with good performance can be obtained by seeking for the one with the smallest $d(\boldsymbol{x}_p,\boldsymbol{r}_q), p=1,2,...,|P'|$.

The method of selecting individuals is showed in algorithm2.

Algorithm2. Selection of individuals

1: $P(t+1)=\varnothing$

2: For $\forall\boldsymbol{x}_p\in P'(t), p=1,2,...,|P'|$ and $\forall\boldsymbol{r}_q\in R_s(t), q=1,2,...,N_r$, calculate $d(\boldsymbol{x}_p,\boldsymbol{r}_q)$ according to formula (5), and form set $D=\{d(\boldsymbol{x}_p,\boldsymbol{r}_q)\,|\,p=1,2,...,|P'|, q=1,2,...,N_r\}$

3: **for** $i=1,2,...,N_p$ **do**

4: Seek for the individual with the smallest $d(\boldsymbol{x}_{p_{\min}},\boldsymbol{r}_{q_{\min}})$ in D

5: $P(t+1)=P(t+1)\cup\{\boldsymbol{x}_{p_{\min}}\}$

6: Delete $d(\boldsymbol{x}_p,\boldsymbol{r}_{q_{\min}}), p=1,2,...,|P'|$ and $d(\boldsymbol{x}_{p_{\min}},\boldsymbol{r}_q), q=1,2,...,N_r$ from D

5: **end for**

As can be observed from the aforementioned analysis, the new population generated in this way has better performances in convergence and distribution than the old. As a result, the Pareto optimal set of the problem can be obtained along with the population's evolution.

3.3 Steps of Proposed Algorithm

Combining Section 3.1 with Section 3.2, the steps of RPEA are described as follows:

Algorithm3. RPEA

1: Set the values of the control parameters, let $t=0$, and initialize the population $P(t)$

2: **while** the termination criterion is not met **do**

3: According to algorithm1, generate a series of reference points, denoted as $R_s(t)$, based on $P(t)$

4: Perform genetic operators on $P(t)$ so as to generate a temporary population $Q(t)$ with the same size as $P(t)$, and let $P'(t)=P(t)\cup Q(t)$

5: According to algorithm2, form $P(t+1)$ by selecting N_p individuals from $P'(t)$, based on $R_s(t)$

6: $t=t+1$

7: **end**

8: Output the optimal solutions

4. EXPERIMENTS

In this section, the performances of RPEA were investigated by comparing it with the other two state-of-the-art MOEAs, NSGA-II [1] and HypE [8]. NSGA-II is a very popular MOEA which adopts the traditional Pareto domination to select individuals. HypE is an indicator-based algorithm which employs a Monte Carlo simulation to approximate the exact hypervolume value, and enables the hypervolume-based search to be easily applied to many-objective optimization. These three methods were applied to solve the following four benchmark optimization problems, i.e., DTLZ1, DTLZ2, DTLZ3, and DTLZ5 [9], and metric IGD [10] were adopted to compare them.

4.1 Parameter Settings

The following parameter settings were adopted by all these methods. The population size was 100 and the size of tournament selection was chosen as 2. The operators for crossover and mutation were simulated binary crossover and polynomial mutation with distribution indexes 20 and 10. The crossover and mutation probabilities were 1.0 and $1/n$, repectively. The termination criterion was that the number of evaluations reaches to the predefined one. For DTLZ1 and DTLZ3, it was set 100 000, and for DTLZ2 and DTLZ5, 30 000.

In RPEA, a large value of α can improve the distribution of reference points, but the time consumption could be very high. A large value of ε is beneficial to improve the efficiency of the evolution, while a small value of ε can diversify the solution set. Basing on our previous experiments, it was appropriate to set α and ε as 0.5 and 0.01, respectively. In HypE, the number of sampling points to estimate hypervolume was set 10 000.

Each algorithm run 30 times on each optimization problem, and the mean values and variances of IGD were calculated. In addition, Mann-Whitney U distribution test was employed to determine whether the metrics obtained by one algorithm has significant differences with those obtained by the other. Finally, the null hypothesis was rejected at a significant level of 0.05.

4.2 Results and Analysis

Table 1 shows metric IGD obtained by different methods, where 'obj' represents the number of objectives contained in an optimization problem, the boldface data are the best among these methods, and those labeled by '†' mean data significantly different from RPEA's.

Table 1. Metric IGD of different methods

Problem	Obj	NSGA-II	HypE	RPEA
DTLZ1	6	4.784E+1 (2.5E+1)†	1.616E+0 (1.0E+0)†	**3.321E-1 (9.4E-2)**
	9	5.652E+1 (4.2E+1)†	1.777E+0 (8.4E-1)†	**5.719E-1 (2.9E-1)**
	12	7.818E+1 (5.0E+1)†	2.522E+0 (9.7E-1)†	**1.363E+0 (7.1E-1)**
DTLZ2	6	1.722E+0 (1.9E-1)†	5.340E-1 (5.4E-2)†	**3.176E-1 (1.3E-2)**
	9	2.071E+0 (1.4E-1)†	6.967E-1 (4.2E-2)†	**5.461E-1 (1.7E-2)**
	12	2.139E+0 (2.0E-1)†	8.508E-1 (6.0E-2)†	**6.964E-1 (3.1E-2)**
DTLZ3	6	1.207E+2 (5.7E+1)†	**3.259E+0 (3.0E+0)**	3.312E+0 (2.1E+0)
	9	1.343E+2 (7.3E+1)†	**3.439E+0 (2.4E+0)†**	4.953E+0 (2.2E+0)
	12	1.405E+2 (8.6E+1)†	**3.874E+0 (1.8E+0)†**	6.874E+0 (2.9E+0)
DTLZ5	6	1.461E-1 (3.2E-2)†	1.565E-1 (7.4E-2)†	**1.043E-1 (4.5E-2)**
	9	4.711E-1 (3.0E-1)†	1.980E-1 (7.7E-2)†	**1.495E-1 (4.3E-2)**
	12	9.118E-1 (2.6E-1)†	2.096E-1 (9.9E-2)†	**1.774E-1 (5.1E-2)**

Based on table 1, (1) when tackling DTLZ1, DTLZ2, and DTLZ5, RPEA has a significantly smaller value of IGD than the other two, therefore has the best performances in convergence and distribution, which validates that the proposed method is very competitive in solving MaOPs; (2) when tackling DTLZ3, RPEA is significantly superior to NSGA-II, but inferior to HypE. The possible reason is that the objective functions have a large range and the individuals close to the boundary have poor performances. By using the reference points generated by these individuals, new superior individuals can hardly be selected in such limited areas. As a result, the population is difficult to evolve. This drawback could potentially be overcome by introducing a new approach to updating the individuals close to the boundary, which will be investigated in future work.

5. CONCLUSIONS

This paper exploited the potential of the reference points in handling MaOPs. The proposed method, RPEA, can mainly be characterized as: (1) adaptively generating a series of reference points with good convergence and distribution based on the evolution of a population; (2) greatly increasing the selection pressure toward the true Pareto front by calculating the distances between the reference points and the individuals in the environment selection process.

The propose method was applied to four benchmark MaOPs, and compared with the other two method to evaluate its performance. The results reveal that RPEA is very competitive to the others in terms of seeking for a solution set with good approximation and distribution in many-objective optimization. The results show that RPEA can achieve a good tradeoff among the convergence and the diversity under a proper setting.

It is worth mentioning that RPEA has been applied only to optimization problems with numerical objectives. Its effectiveness in other optimization problems, especially in engineering optimization, should further be confirmed. In addition, RPEA adopts a key parameter, ε, which will affect its performance. If appropriate methods are employed to adaptively adjust the value of ε during the evolution, the performance of RPEA will further be improved, which is also our further research work.

6. ACKNOWLEDGMENTS

This work was jointly supported by the National Science Foundation of China with granted number 61375067, 61105063, and 61075061.

7. REFERENCES

[1] Deb, K., Pratap, A., Agarwal, A. and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

[2] Wierzbicki, A. P. 1980. The use of reference objectives in multiobjective optimization. *Multiple Criteria Decision Making Theory and Application*. Springer-Verlag, Berlin, 468-486.

[3] Deb, K. and Sundar, J. 2006. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, (Seattle, WA, USA, 2006), ACM, 635-642.

[4] Mohammadi, A., Omidvar. M. N. and Li, X. 2012. Reference point based multi-objective optimization through decomposition. In *Proceedings of the Congress of Evolutionary Computation*, (Brisbane, QLD, 2012), IEEE, 1150-1157.

[5] Figueira, J. R., Liefooghe, A., Talbi, E. G. and Wierzbicki, A. P. 2010. A parallel multiple reference point approach for multi-objective optimization. *European Journal of Operational Research*, 205(2), 390-400.

[6] Deb, K. and Jain, H. 2013. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 99, 1.

[7] Wang, R., Purshouse, R. C. and Fleming, P. J. 2012. Preference-based solution selection algorithm for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(1), 20-34.

[8] Bader, J. and Zitzler, E. 2011. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1), 45-76.

[9] Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. 2002. Scalable test problems for evolutionary multiobjective optimization. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, (Hawaii, US, 2002), IEEE Computational Intelligence Society, 825-830.

[10] Zhang, Q. F., Zhou, A. M., Zhao, S. Z., Suganthan, P. N., Liu, W. D. and Tiwari, S. 2008. *Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition*. Technical Report. University of Essex, Colchester, UK and Nanyang Technological University, Singapore.