A Simple Pareto Adaptive $\epsilon$-Domination Differential Evolution Algorithm for Multi-Objective Optimization

Yan Jingfeng$^{1,*}$, Li Meilian$^1$, Xu Zhijie$^1$ and Xu Jin$^2$

$^1$School of Information Engineering, Xuchang University, Xuchang, Henan, 461000, China, P.R. China
$^2$School of Computer, China University of Geosciences, Wuhan, Hubei, 430074, China, P.R. China

Abstract: The two purposes of solving the multi-objective optimization problems are to get solutions close to the true Pareto front as much as possible and to obtain promising diversity. To meet these two demands, a new method is proposed in this paper, which has these characteristics: 1) it adopts the orthogonal design method with quantization technology to generate initial population whose individuals are scattered uniformly over the target search space. 2) it is based on an adaptive $\epsilon$ concept to obtain a good distribution along the true Pareto-optimal solutions. Experiments on five benchmark problems with different features indicate that the proposed method works well not only in diversity, but also in convergence when compared to other evolutionary algorithms.

Keywords: Multi-objective optimization, differential evolution, adaptive $\epsilon$ domination, pareto optimal.

1. INTRODUCTION

In the natural sciences and social sciences, we hope to maximize efficiency and minimize costs. Which is essentially a multi-objective optimization problem [1]. As each objective restricts each other through decision variables in multi-objective optimization problems, it will result in the loss of other target's performance to over-optimize one objective. Therefore, it is difficult to evaluate the pros and cons of the solutions of multi-objective optimization problem. There is no unique global optimal solution in multi-objective problem, in stead, it is a collection of the optimal solution which is commonly known as the Pareto optimal solution set. It is different from single objective optimization problem which is clearly defined and has just single optimal solution. Elements in the Pareto optimal solution set are non-domination and not comparable, that is to say, when considering all targets, there does not exist a solution which is better than these solutions. The main task of the multi-objective algorithm is to find solutions with good convergence and diversity which meet the requirements of the representative.

In last decades, a lot of algorithms were introduced to address multi-objective optimization problem, for instance, aggregation approaches, VEGA algorithm [2], the lexicographic ordering, $\epsilon$-constrains method, the target-vector method, NSGA [3], MOGA [4], NPGA [5]. In the late 1990s, some improved algorithms were come out, such as: PAES [6], SPEA [7], NSGA-II [8], NPGA [9]. Researches on multi-objective genetic algorithm design and theoretical in domestic also showed the situation to go out in the ascendant. Yuping Wang [10] and Sanyou Zeng [11] give a new method for solving multi-objective optimization respectively by combining orthogonal design with uniform design into genetic algorithm, which accelerate the convergence speed. Shihua Guan [12] introduces an Augmented Lagrangian multi-objective collaborative algorithm based on the $\epsilon$-constraint method. Chuan Shi [13] proposes a quick multi-objective evolution algorithm based on domination tree, it assigns fitness through domination tree which results in a less comparison among individuals. Gong etc [14] apply two-level orthogonal crossover operator to DE and select optimal individuals using statistical optimization which lead to a better robust. Maoguo Gong [15] generalizes the current trend of research on multi-objective optimization and then put forward their own views on the further development of multi-objective optimization.

Although there have been many evolution algorithms to solve multi-objective optimization problems, it is still a challenge to design an efficient and robust algorithm. To this end, the authors introduce a differential evolution algorithm associating with $\epsilon$ domination to solve multi-objective optimization, which termed Pa$\epsilon$-ODEMO. The algorithm has following characteristics:

1. applying orthogonal design method to generate the initial population, which not only decreases the time consumption, but also makes the initial population of points scattered uniformly over the feasible solution space.

2. using an Archive population to retain the obtained non-dominated solutions. What's more, an adaptive $\epsilon$ way is adopted to maintain the diversity and distribution of the archive population dynamically.
Finally, experiments on five benchmark problems of diverse complexities have shown that the new approach is able to achieve comparable result in terms of convergence and diversity metrics when compared with several other state-of-the-art evolutionary algorithms.

The paper is organized in the following manner. Section 2 introduces some basic knowledge about multi-objective optimization and differential evolution algorithm. Section 3 presents the core idea of our algorithm Pa-ODEMO. Section 4 summarizes some experimental results on continuous problems. Finally, experiments on five benchmark problems of different characteristics are made to achieve comparable result in terms of convergence and diversity metrics when compared with several other state-of-the-art evolutionary algorithms.

2. MULTI-OBJECTIVE DIFFERENTIAL EVOLUTIONARY ALGORITHM

In this section, we mainly focus on introducing some prerequisite knowledge about multi-objective optimization and differential evolution algorithm. They are the foundation of Pa-ODEMO.

2.1. Relative Description on Multi-objective Optimization

Definition 1: (multi-objective optimization problem): without loss of the generality, a MOP with a set of n decision variables, a set of m constrains and a set of k objective functions can be described as following:

\[
\begin{align*}
\text{minimize:} & \quad \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \\
\text{subject to:} & \quad \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), \ldots, e_m(\mathbf{x})) \geq 0 \\
\text{where:} & \quad \mathbf{x} = (x_1, x_2, \ldots, x_n) \in X \\
& \quad \mathbf{y} = (y_1, y_2, \ldots, y_k) \in Y
\end{align*}
\]

where \( \mathbf{x} \) is the decision vector, \( \mathbf{y} \) is the objective vector. \( X \) denotes as the decision space, while \( Y \) means the objective space. Generally, for each decision vector, it satisfies the constrain functions.

Definition 2: (pareto dominance) assuming objective vectors \( \mathbf{x} = (x_1, \ldots, x_n) \) and \( \mathbf{y} = (y_1, \ldots, y_k) \) are two feasible solutions. if and only if

\[ i \neq 1, \ldots, k \quad x_i \leq y_i \quad \text{and} \quad i = 1, \ldots, k \quad x_i < y_i \]  

which is said \( \mathbf{x} \) Pareto dominate \( \mathbf{y} \). denotes: \( \mathbf{x} \prec \mathbf{y} \)

Definition 3: (\( \epsilon \) dominance) a given objective vector \( \mathbf{x} = (x_1, \ldots, x_n) \) is said to \( \epsilon \) dominate another objective vector \( \mathbf{y} = (y_1, \ldots, y_k) \), if and only if

\[ \forall i \in 1, \ldots, k \quad (1-\epsilon)x_i \leq y_i \]  

which denotes: \( \mathbf{x} \prec_{\epsilon} \mathbf{y} \)

Definition 4: (pareto optimal) given a solution \( \mathbf{x} \) which is pareto optimal, if and only if

\[ \nexists \quad \mathbf{x} \in X \quad \text{makes} \quad \mathbf{x} \prec \mathbf{x}^* \]  

Definition 5: (pareto front) the pareto font is the appearance formed by all pareto-optimal solutions when mapped to the objective vector space. its mathematical description is as following:

\[ \text{POF} = \{ f(x) = (f_1(x), \ldots, f_k(x)) | x \in X \} \]  

2.2. Differential Evolution Algorithm

DE is a random-based heuristic search algorithm, it is a kind of evolution algorithm based on real-coded which is proven to be an effective technique to solve complex optimization problems. It has been developed to be a mature algorithm and its framework can be described as following: (1) Parameter assignment and generating the initial evolution population, set the evolution generation time=0;

(2) Calculate the fitness of the initial population;

(3) Stopping the algorithm and output the result if the stopping condition is met, else jump to step (4);

(4) Apply crossover operator, mutation operator and selection operator to the population, and then generate the next generation;

(5) Calculate the fitness of the population;

(6) Let the generations \( t \rightarrow t+1 \), then jump to step (3);

What is necessary to point out is that our algorithm employs a hybrid selection mechanism in which a random selection and an elitist selection are interleaved. To be specific, in the early evolution, individuals in Archive population have big difference with the pareto optimal solutions, so they do little help to guide the evolution, at this moment, the algorithm selects individuals from evolution population directly. However, with the progressing of evolution, individuals in Archive population will be closer and closer to the pareto optimal set. At this time, individuals are chosen from Archive population randomly, namely:

\[ \text{individual} = \{ \begin{array}{ll}
\text{random} & \text{eval} < (\lambda \times \text{Max}_\text{eval}) \\
\text{elitist} & \text{otherwise}
\end{array} \]  

3. PAe-ODEMO AND ITS CHARACTERISTICS

In this section, we will present the proposed algorithm Pa-ODEMO in detail. Compared with traditional evolution algorithm for multi-object, it has many improvements and characteristic.

3.1. Initial Population With Orthogonal Design

In traditional DE, the initial population is generated randomly, which often lead to an uneven distribution, and then, decrease the algorithm’s ability to use the initial population. While the initial population generated by orthogonal design method [16] can have a better distribution and diversity along the decision vector space. Orthogonal initial population can not only improve utilization capacity of the initial population, but also accelerate the convergence speed. There are two mainly steps to generate the initial orthogonal population:

(1) generating Orthogonal array [17], it’s a kind of Latin array, using \( L_n(Q^C) \) to represents an array with Q level;
(2) Quantization of the search space [18], assuming a variable \( x_i \) has a boundary of \([l,u]\), and then quantize it into \( Q \) level as following:

\[
a_i = l + (i-1)(u-l)/(Q-1) \quad i = 1,2,\ldots,Q
\]

(7)

3.2. Adaptive \( \varepsilon \) to Maintain Diversity of the Archive

In the procedure of updating population in this algorithm, whenever a new offspring is produced, it should be compared not only with its parents, but also with the individuals in archive population. If the new offspring is better than some individuals in archive, then just delete those individuals who are dominated; if the new offspring and all individuals in archive are not dominated mutually, then add it to the archive; if it is dominated by someone individual in archive, then discard the new one. In this case, as the running of evolution, the size of archive population will be larger and larger, which will make it hard to store, at mean while, the distribution of archive will be getting worse and worse (individuals within a region have a high density, but other regions a small density). In order to obtain not only a better distribution, but also reducing the storage space, the algorithm uses adaptive \( \varepsilon \) grid to maintain the diversity of archive.

When the size of archive population reaches a predefined value, we need to generate the grid and map all individuals in archive population to the grid, and then using grid method to update the archive population. In order to store these mapped individuals, we assign an identify array to each individual, where \( r \) represents the number of objectives. The formula to calculate the \( B \) is shown as following:

\[
B_i = (\text{int})(f_{i} - f_{i}^{\text{min}})/\varepsilon_i \quad i = 1,\ldots,r
\]

(8)

Where \( f_{i}^{\text{min}} \) represents the minimum of the \( i \)th objective, the individual with the minimum fitness is from archive, it varies as the running of evolution. The \( E_i \) means the tolerance of the \( i \)th objective, with \( E_i \) we can map each individual into the grid easily.

In order to generate the grid we need to calculate the corresponding \( E_i \). Without loss of generality, assume that one objective function has a range of \( 1 \leq f_i \leq K \), then for an optimization problem with \( r \) objective, it will produce a number of \( r \) grid which can accommodate \((K-1/\varepsilon)^r\) non-dominated solutions. For an evolution population with \( NP \) size, the algorithm hopes to produce \( NP \) non-dominated solutions, so it will be met:

\[
NP = (K-1/\varepsilon)^{r-1}
\]

(9)

And for this reason, we can calculate each \( E_i \) easily. The reason why the \( E_i \) is adaptive is that in most multi-objective optimization problems, the range of an objective is unknown at the beginning, but it is essential to know its range when calculate \( E_i \). Here, the maximum and minimum for a certain objective function are obtained from the archive population, because the algorithm thinks that individuals in archive are optimal, although they are just temporary optimal. As the evolution progresses, the maximum and minimum of a certain objective function may change, so that we can adjusted the \( E_i \) through the changing of its range, thereby adjust the location of individual mapped to the grid.

Algorithm 1 shows the procedure of our algorithm update archive population. When a new offspring is generated, we use (8) to calculate its identify array \( B \). Obviously, if \( B[i] < 0 \), the minimum of \( i \)th objective will be change and the offspring is out of the original grid. And then, we should adjust the grid to accommodate it. However, to avoid adjusting grid too frequently, we just regenerate grid when \( B[i] < 3 \), and then readjust the position of each individual in the grid.

4. SIMULATION RESULT

In this section, we present some benchmark functions and then conduct some simulations to verify the convergence and diversity of our algorithm. What’s more, we also analyze the results.

4.1. Test Case

In order to test the pareto front of the multi-objective problems with various features, for instance convex, concave, discrete, uniform distribution, we have adopted some standard test functions below:

Case 1:

\[
\begin{align*}
\begin{cases}
\quad f_1(x) = x_1 \\
\quad f_2(x) = g(x)[1-(1-x_1)/g(x)]^{0.5} \\
\quad g(x) = 1+9\sum_{i=2}^{n} x_i/(n-1)
\end{cases}
\end{align*}
\]

In ZDT1, the size of decision vector \( n \) is 30, for each variable \( x_i \in [0,1] \), it has a convex pareto optimal front.

Case 2:

\[
\begin{align*}
\begin{cases}
\quad f_1(x) = x_1 \\
\quad f_2(x) = g(x)[1-(1-x_1)/g(x)]^{2} \\
\quad g(x) = 1+9\sum_{i=2}^{n} x_i/(n-1)
\end{cases}
\end{align*}
\]

In ZDT2, the size of decision maker \( n \) is 30, for each variable \( x_i \in [0,1] \), it has a concave pareto optimal front.
Case 3: ZDT3
\[
\begin{align*}
    f_1(x) &= x_1 \\
    f_2(x) &= g(x) [1 - (1 - \sqrt{x_1 / g(x)} - x_1 \sin(10\pi x_1) / g(x))] \\
    g(x) &= 1 + 9 \left( \sum_{n=2}^{n} x_n \right) / (n-1)
\end{align*}
\]

In ZDT3, the size of decision maker n is 30, for each variable \(x_i \in [0,1]\), its pareto optimal front is discrete and concave.

Case 4: ZDT6
\[
\begin{align*}
    f_1(x) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\
    f_2(x) &= g(x) [1 - (f_1(x) / g(x))^2] \\
    g(x) &= 1 + 9 \left( \sum_{n=2}^{n} x_n \right) / (n-1)^{0.25}
\end{align*}
\]

In ZDT6, the size of decision maker n is 10, for each variable \(x_i \in [0,1]\), it has a concave nonuniform distribution of pareto optimal front.

Case 5: DTLZ1
\[
\begin{align*}
    f_1(x) &= 0.5x_1g(x) + 0.5x_1(1 + g(x)) \\
    f_2(x) &= 0.5x_1(1 - x_1)(1 + g(x)) \\
    f_3(x) &= 0.5x_1(1 - x_1)(1 + g(x)) \\
    g(x) &= 100[\sum_{i=1}^{n} x_i + \sum_{i=3}^{n} (x_i - 0.5)^{0.5} \cos(20\pi(x_i - 0.5))] \\
\end{align*}
\]

In ZDT6, the size of decision maker n is 12, for each variable \(x_i \in [0,1]\), it is a three objective optimization problem.

4.2. Evaluation Indicators

(1) Convergence \(\gamma\), denotes as . It was proposed by Van Veldhuizen and Lamont in 1998, used to represent approximation degree of the obtained optimal set \(Q\) and the real pareto front \(PF\), which is calculated as following:
\[
\gamma = \sum_{i=1}^{n} d_i / |Q|
\]

Where \(d_i\) means the smallest Euclidean distance between individual \(i\) in \(Q\) and individuals in \(PF\), the smaller value of \(\gamma\), the better the algorithm performs. It’s one of principle indicators

(2) \(\Delta\) metric. Deb etc, proposed the indicator in 2002, which is used to evaluate the distribution along the pareto front, its formula is described as following:
\[
\Delta = (d_1 + d_1 + \sum_{i=1}^{n} d_i - \bar{d}) / (d_1 + d_1 + |Q - 1|) \bar{d}
\]

In the formula above, \(d\) means the Euclidean distance of all \(d_1\), \(d_1\) and \(d_1\) represents the distance between the boundary solutions of \(Q\) and pareto front which are mainly used in evaluating bi-objective optimization problem. The smaller of \(\Delta\) value, the better the algorithm performs, especially when \(\bar{d} = d_1\), \(\Delta = 0\), at this time the solution obtained by this algorithm distribute uniformed along the Pareto front.

However, this indicator has its defects through analysis. For instance, it is mainly used to evaluate bi-objective optimization which is of limitation to a certain extent. What’s more, if there are only two boundary solutions in \(Q\), \(\bar{d}\) is the average of all \(d_1\), \(d_1\) and \(d_1\) represents the distance between the boundary solutions of \(Q\) and pareto front which are mainly used in evaluating bi-objective optimization problem. The smaller of \(\Delta\) value, the better the algorithm performs, especially when \(\bar{d} = d_1\), \(\Delta = 0\), at this time the solution obtained by this algorithm distribute uniformed along the Pareto front.

Nevertheless, it is an validate indicator to evaluate performance of most algorithms.
4.3. Experiment Results and Analysis

The experiment is based VC++6.0 platform, each algorithm runs 20 times independently, and then compute its average. In this algorithm, evolution parameters are set as following: the size of population $NP=100$, the max fitness evaluations $Maxeval=25000$, scale factor $F=0.5$, selection control paramete $\lambda=0.1$, the index of orthogonal design $J=2$. For ZDT1, ZDT2, ZDT3, $Q=29$; for ZDT6 and DTLZ1, $Q=21$, the size of archive size $NF=100$ and crossover probability $CR=0.9$. In this paper, Pae-ODEMO is compared with several other state-of-the-art evolution algorithms: NSGA-II, SPEA2, INSGA-II [19], AEPSO [20], NPCA* [21], PBFO [22], DEMO [23], $\varepsilon$-DEMO [24], to verify its performance.

Seen from Table 1, the Pae-ODEMO does well in convergence. it has the best convergence value in all test cases. On ZDT6 and DTLZ1, NSGA-II and SPEA2 is inclined to local pareto front. NSGA-II use crowd distance to sort populations, while INSGA-II is based on a loop crowd distance concept which improves the convergence compared to NSGA-II. DEMO performs badly in convergence, but its convergence improves obviously after adopting dominance method. Nevertheless, its performance in convergence is far from that of Pae-ODEMO in all cases. The experimental results show that the algorithm can approach to the true pareto front well.

Seen from Table 2, the algorithm also achieves good result in diversity. On ZDT1, it ranks only second only to INSGA-II and on DTLZ1, its result is Intermediate. Apart from this, the algorithm performs best in the rest cases. In aspect of maintaining diversity, equal intervals grid is adopted in our algorithm which makes sure that the distance of any two neighbor points fixed in a small range, as a result.
Fig. (2). Search trace for minimize 30-D ZDT2, it can get two boundary solutions easily.

Fig. (3). Search trace for minimize 30-D ZDT3, it is discrete.

Fig. (4). Search trace for minimize 10-D ZDT6.
all non-dominance solutions are scattered uniformly along the pareto front. However, For some discontinuous problems, for example ZDT3, it decomposes the solution space into several parts, at this time, the equal interval grid can just make sure a good distribution in its continuous part. to this end, if the algorithm enlarge its $\varepsilon$ value, it will result in a smaller value. Experiments shows that if doubling the $\varepsilon$ value, its can reach to 0.4. Obviously, the algorithm achieve a nice result both in convergence and diversity. However, Zitzler etc [25] point out that an algorithm’s performance can’t just be evaluated by numeric indicator. Therefore, in this paper, Fig. (1-5) of solutions run on the Matlab platform are shown to give a visual representation. Fig. (1-5) is the result of Pa-ODEMO run for once.

CONCLUSION

An adaptive $\varepsilon$ dominance based orthogonal differential evolution algorithm for multi-objective optimization is proposed in this paper. The main improvement of this new approach is that it introduces an adaptive $\varepsilon$ dominance method to update the archive population which can maintain the diversity and distribution of population in an adaptive way.

Experiments on five benchmark functions and comparison with several classical algorithm: NSGA-II, SPEA2, INSGA-II, AEPSO, NPCA and $\varepsilon$-DEMO indicate that the new algorithm Pa-ODEMO does well in both convergence and diversity. What’s more, it’s robust. However, seen from the figures above, it is obvious when the slope of tangent tends to 0 or 1, the points is few and scattered. Our later work will devote to adjusting the density of $\varepsilon$ grid adaptively, that is to say, when the slope of tangent tends to 0 or 1, the value of $\varepsilon$ is small which expresses as a higher density of grid while the $\varepsilon$ grid will be few and scattered at other parts.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.
A Simple Pareto Adaptive ε-Domination Differential


