ISBN 978-981-11-7861-0
Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)
Bangkok, 28-30 June, 2018, pp.713-717
doi: 10.18178/wcse.2018.06.117

A Novel Objective Reduction Algorithm Using Objective Sampling for Many-Objective Optimization Problems

Minghan Li [1], Jingxuan Wei [1 +] and Long Zhao [1]

[1] Department of Computer Science, Xidian University, Xi'an, China

Abstract. In the field of science and engineering, many problems are Many-objective Optimization Problems (MaOPs), which have more than three objectives. The main difficulty of MaOPs is the true Pareto front is hard to get due to the low selection pressure. However, for some MaOPs, we can reduce the number of objectives to get the non-redundant objectives. In this paper, a novel fast objective reduction algorithm is proposed. Different from other objective reduction algorithms, this algorithm uses a sampling method to get the relationships between objectives by calculating objectives' improvements. Then, a fast procedure is used to omit the redundant objectives. Finally, experiments show that the proposed algorithm is effective.

Keywords: many-objective optimization, objective reduction, soft computing, intelligence computation

1. Introduction

Many real-world optimization problems are Multi-objective Optimization Problems (MOPs), and many MOPs have more than three objectives, which are called Many-objective Optimization Problems (MaOPs). Most of the proposed Multi-Objective Evolutionary Algorithms (MOEAs) use the Pareto-dominance relation to compare solutions of the population [1]-[3]. However, their search ability is severely deteriorated when the number of objectives increases, especially in solving MaOPs [4]. The main challenge is, when the number of objectives increases, the proportion of non-dominated solutions increases rapidly. It means that almost all solutions become non-dominated. This makes the search ability very poor. Nevertheless, the recently proposed many-objective algorithms may not effectively handle MaOPs with more than over 15 objectives [5].

Fortunately, many real-world problems have redundant objectives. That is to say, we can try to remove the redundant objectives to improve the search ability. In recent years, much work has been done to remove the redundant objectives, e.g., [6]-[9]. However, these existing algorithms have some limitations, such as needing non-dominated solutions by using an evolutionary algorithm and high computational cost.

In order to address the limitations, this paper proposes a new objective reduction algorithm. By sampling the objectives and using a fast reduction selection method, non-redundant objectives are obtained. Experiments show that the proposed algorithm works efficiently and successfully.

2. Problem Definition

2.1. Problem Definition

Many-objective Optimization Problems

Without loss of generality, a multi-objective optimization problem can be described as (1) [10].

$$\begin{cases} \min f(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \\ \qquad s.t. \ x \in \Omega \end{cases} \tag{1}$$

+ Corresponding author. Tel.: +8613891994573.
E-mail address: wjx@xidian.edu.cn

where x = $(x_1,x_2,...,x_n)$ is the decision vector, Ω is the search space, $f_i(x)$ is the i-th objective function in the objective space. We call the problem many-objective optimization problem when the number of objectives is more than three.

Pareto dominance

A solution x = $(x_1,x_2,...,x_n)$ is said to dominate (denoted by ≺) another solution y = $(y_1,y_2,...,y_n)$ if and only if f(x) is partially less than f(y). That means, ∀m ∈ $\{1,...,M\}$, we have $f_m(x) \le f_m(y)$ and ∃m ∈ $\{1,...,M\}$, where $f_m(x) < f_m(y)$.

Pareto optimal solution

A solution $x = (x_1,x_2,...,x_n)$ is said to be optimal solution if and only if there is no $y = (y_1,y_2,...,y_n)$ that y dominates x with respect to solution space.

Pareto front

Given MOP f(x) and its Pareto optimal solution set P, the Pareto front is PF={f(x),x∈P}.

2.2. Related Work

Deb and Saxena propose PCA-NSGA-II [6]. The eigenvalues and corresponding eigenvectors of the correlation matrix are used to identity the most important conflicting objectives which should not be removed. This algorithm relies on the non-dominated solution set obtained by MOEA to compute the correlation matrix.

Brockhoff and Zitzler propose a new definition of conflict which aims at finding a subset of the original objectives [7]. A greedy and an exact algorithm were proposed to solve the k-EMOSS and the δ-MOSS problems. However, its time complexity is quite high, which limits its practical applications.

Singh et al. propose the PCSEA [8]. The authors use boundaries of the Pareto front called corner solutions by which the true dimensionality of the Pareto front can be predicted. However, it still needs to get the non-dominated solutions and takes a high computational cost.

Jaimes et al. proposed an unsupervised feature selection technique to remove the redundant objectives [9]. This method also uses a correlation matrix of a set of non-dominant solutions to measure the conflict between each pair of objectives. The limitations of this algorithm are this algorithm needs to artificially specify a neighborhood size q for clustering and it also relies on non-dominated solutions.

3. Proposed Reduction Algorithm

First, we sample the objectives and obtain M (M is the number of objectives) data points that represent objectives' gradient information. Second, we analyze the M data points and get the non-redundant objectives.

3.1. Sample the Points

This paper presents a novel approach to obtain the relationships between objectives by sampling them. The procedure to obtain data points is as follows: first, initialize N decision vectors randomly distributed in objective space. Second, calculate objective values by above N decision vectors and obtain objective values matrix A,

$$\mathrm{A} = \begin{bmatrix} f_1(x^1) & \cdots & f_1(x^N) \\ \vdots & \ddots & \vdots \\ f_M(x^1) & \cdots & f_M(x^N) \end{bmatrix} \tag{2}$$

where, each row of matrix A represents N values of each objective and each column of matrix represents M objectives' values of one sampling decision vector x. Third, give a tiny increment to each decision vector x in every dimension, $x_{new} = (x_1 + \varepsilon, x_2 + \varepsilon, ..., x_n + \varepsilon)$. We calculate objective values by above N increased sampling decision vectors and obtain matrix B of objective values.

$$\mathrm{B} = \begin{bmatrix} f_1({x_{new}}^1) & \cdots & f_1({x_{new}}^N) \\ \vdots & \ddots & \vdots \\ f_M({x_{new}}^1) & \cdots & f_M({x_{new}}^N) \end{bmatrix} \tag{3}$$

Last, we obtain matrix C by $B - A$. To each value C_{ij} in matrix C, if $C_{ij} > 0$, then set it to 1; if $C_{ij} < 0$, then set it to -1. Each row of matrix C represents each objective's gradient information in decision space and we can take it as a high-dimensional point, which can approximately represent the objective. If the gradient trends of two objectives are very similar, then we can delete an objective. The procedure of generating M points that approximately represent M objectives is given in Algorithm 1.

3.2. Pick Out the Non-Redundant Objectives

After sampling the objectives, we have M points that represent objectives. In the following, this algorithm proposes an approach to omit the redundant objectives. In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different [11]. Let P_i represent one point corresponding to one objective and P_j represent another point corresponding to another objective. Let $Dis_{P_{ij}}$ represent the Hamming distance between P_i and P_j. It is not hard to see if $Dis_{P_{ij}}$ is relatively low, then the objective corresponding to P_i might be a redundant objective. Based on the analysis above, there should be a parameter R to judge whether the objective corresponding to P_i is a redundant objective. This parameter R is shown in (4),

$$\mathrm{R} = \frac{Dis_{P_{ij}}}{N} \tag{4}$$

where N denotes the dimension of the point and it is also the times of sampling. If R is lower than the threshold, the objective corresponding to P_i is viewed as a redundant objective and it needs to be removed, meanwhile, P_i goes to the next point P_{i+1} instead of comparing with all other points, which reduces the time complexity. The removal process is repeated until all points are calculated and examined, then we obtain the non-redundant objectives. The process of picking out non-redundant objectives is shown in Algorithm 2. The main loop of the proposed algorithm is described in Algorithm 3.

Algorithm 1: Obtaining data points

Input : decision matrix X of N decision vectors;
Output: data points Points;

A ← calculateFuncVal(X);
X_{new} ← tinyIncrement(X);
B ← calculateFuncVal(X_{new});
C ← B − A;
C ← normalization(C);
for *each* row ∈ C **do**
    Points.add(row);
end

Algorithm 2: Obtaining non-redundant objectives

Input : data points Points;
objectives Objs;
Output: non-redundant objectives NObjs;

Tem ← Points;
for *each* $P_i \in Points$ **do**
    for *each* $P_j \in Tem$ **do**
        if P_i is not P_j **then**
            $Dis_{P_{ij}}$ ← calHammingDis(P_i, P_j);
            R ← calRvalue($Dis_{P_{ij}}$);
            if R ≤ ratio **then**
                Objs ← deleteTheRedun(Objs, P_i);
                Tem ← deleteP(Tem, P_i);
                break;
            end;
        end;
    end;
    NObjs ← Objs;
end

Algorithm 3: Main Loop

Input : decision matrix X of N decision vectors;
objectives Objs;
Output: non-redundant objectives NObjs;

Points ← obtainDataPoints(X);
NObjs ← obtainNonRedundant(Points, Objs);

4. Experiment Design

We adopt test problems DTLZ2(M) [12] and DTLZ5(I,M) [6] in the experiments, where M is the original number of objectives and I represents the actual dimensionality of the Pareto front. The first problem has no redundant objectives, and this can test whether the proposed algorithm could omit objectives mistakenly. The second problem is a redundant problem. We use it to test the accuracy of obtaining non-redundant objectives. In order to show the improvement of MaOPs after obtaining the non-redundant objectives, we integrate the proposed objective reduction algorithm into NSGA-II [13] and NSGA-III [5], and then we compare the evolution performance of DTLZ5(I,M) before-and-after omitting the redundant objectives.

For obtaining points that approximately represent the objectives, we set the number of sampling points for each objective to 2000, and the tiny increment ε to each decision vector x is $\mathrm{bound}/2000$ (where bound is the range of decision variables and if range is infinite, set ε to 0.01). A large number of experiments show that the parameter threshold $R = 0.06$ is the best for most problems, which determines whether the objective is viewed as a redundant objective. If $R < 0.06$, then the objective is viewed as a redundant objective and it is removed. As to compare the performance before-and-after omitting the redundant objectives, in NSGA-II and NSGA-III, the population size is 100, the crossover probability is 0.9, the mutation probability is $1/n$ (where n is the number of decision variables), and the number of evaluations is 200000.

5. Results

5.1. DTLZ2(M) and DTLZ5 (I,M)

The true non-redundant objective sets of DTLZ2(M) and DTLZ5(I,M) are shown in Table 1. After 20 times experiments on each DTLZ2(M) and DTLZ5(I,M) test problem, the success rates of the proposed algorithm which represent the accuracy of obtaining the true non-redundant objectives are shown in Table 1.

Table 1: The Result of Test Problems

Problems	Non-redundant objective set	Success Rate
DTLZ2(5)	$\{f_1,f_2,f_3,f_4,f_5\}$	20/20
DTLZ2(7)	$\{f_1,f_2,\ldots,f_6,f_7\}$	20/20
DTLZ2(10)	$\{f_1,f_2,\ldots,f_9,f_{10}\}$	20/20
DTLZ5(3,5)	$\{f_k,f_4,f_5\}\ k \in \{1,2,3\}$	20/20
DTLZ5(4,7)	$\{f_k,f_5,f_6,f_7\}\ k \in \{1,2,3,4\}$	20/20
DTLZ5(5,10)	$\{f_k,f_7,\ldots,f_{10}\}\ k \in \{1,2,\ldots,6\}$	20/20
DTLZ5(7,15)	$\{f_k,f_{10},\ldots,f_{15}\}\ k \in \{1,2,\ldots,9\}$	20/20

As shown in Table 1, none of objectives is omitted for DTLZ2(M) test problems, and the results of obtaining non-redundant objectives for DTLZ5(I,M) are perfectly accurate. The results show that the proposed algorithm does not omit objectives incorrectly for non-redundant problems and omits objectives accurately for redundant problems. In the following, the proposed objective reduction algorithm is integrated into NSGA-II, and an experiment of comparing the performance before-and-after omitting redundant objectives is conducted.

5.2. Integrating the Proposed Algorithm

In this section, NSGA-II and NSGA-III are firstly used to solve the many-objective optimization problem DTLZ5(5,10), which has redundant objectives. Then the proposed objective reduction algorithm is integrated into NSGA-II and used to solve DTLZ5(5,10) again. By doing so, we will see whether the performance will be better by omitting non-redundant objectives, and in turn, the contribution of this work is demonstrated. The Inverted Generational Distance (IGD) that is a variant of GD [14] is used for above three experiments. The smaller the IGD value is, the better the result is. The average IGD of above three experiments after execution for 2000 times is shown in Table 2.

Table 2: Performance Before-and-after integrating the algorithm

MOEA	Integrated?	IGD
NSGA-II	No	120.4
NSGA-III	No	0.417
NSGA-II	Yes	0.393

From Table 2, we see that before integrating the proposed objective reduction algorithm, traditional MOEA as NSGA-II will deteriorate the search ability to converge towards the Pareto front, and its IGD is 120.4. While after integrating the proposed algorithm, NSGA-II's IGD is greatly reduced to 0.393, which is even better than NSGA-III's result. So it proves the objective reduction algorithm is a good approach to improve the search ability for MaOPs.

6. Conclusions and Future Work

The goal of this paper was to investigate a novel objective reduction algorithm for Many-objective Optimization Problems. Experimental results showed that the goal was successfully achieved by developing a new sampling approach to get the relationships between objectives, and a procedure to pick out non-redundant objectives. With the non-redundant objectives, the selection pressure of population with MOEAs is higher and we can ease the search processes. In the future, we consider using other approaches to pick out the non-redundant objectives from the sampling points. It is also interesting to use other techniques to get the relationships between objectives.

7. Acknowledgment

This work is supported by the national science foundation of China (jj0500120310).

8. References

[1] S. Bechikh, M. Elarbi, and L. B. Said, "Many-objective optimization using evolutionary algorithms: A survey," *in Recent Advances in Evolutionary Multi-objective Optimization.* Springer, 2017, 105–137.

[2] Maltese J, Ombuki-Berman B M, Engelbrecht A P, "A scalability study of many-objective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, 22(1), 2018, 79-96.

[3] Bao C, Xu L, Goodman E D, et al. A novel non-dominated sorting algorithm for evolutionary multi-objective optimization[J]. *Journal of Computational Science*, 2017, 23: 31-43.

[4] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," *in Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on. IEEE, 2008*, pp. 2419–2426.

[5] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints." *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, 2014, 577–601.

[6] K. Deb and D. Saxena, "Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," *in Proceedings of the World Congress on Computational Intelligence (WCCI-2006),* 2006, pp. 3352–3360.

[7] D. Brockhoff and E. Zitzler, "Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization," *in Parallel Problem Solving from Nature-PPSN IX*. Springer, 2006, 533–542.

[8] H. K. Singh, A. Isaacs, and T. Ray, "A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, 2011, 539–556.

[9] A. L´opez Jaimes, C. A. Coello Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," *in Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 673–680.

[10] K. Deb, Multi-objective optimization using evolutionary algorithms. *John Wiley & Sons*, 2001, vol. 16.

[11] R. W. Hamming, "Error detecting and error correcting codes," *Bell Labs Technical Journal*, 1950, vol. 29, no. 2, 147–160.

[12] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," *in Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1. IEEE, 2002, pp. 825–830.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, 2002, vol. 6, no. 2, 182–197.

[14] D. A. Van Veldhuizen and G. B. Lamont, "On measuring multiobjective evolutionary algorithm performance," *in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1. IEEE, 2000, pp. 204–211.