A Symmetric Points Search and Variable Grouping Method for Large-scale Multi-objective Optimization

1st Dandan Tang, Yuping Wang*, Xiangjuan Wu
School of Computer Science and Technology
Xidian University, Xi'an, 710071, China
ddtang@stu.xidian.edu.cn
ywang@xidian.edu.cn*, wu870803@163.com

2nd Yiu-ming Cheung
Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong, China
ymcheung@comp.hkbu.edu.hk

***Abstract*—In this paper, we propose a new method for large scale multi-objective optimization based on symmetric points search and variable grouping, named SSVG. The main idea is to use variable grouping scheme first to divide the original decision space into several subspaces. In each subspace, the symmetric points of the points in population form some potential search directions. Using the search directions, the possibility of finding the optimal solutions will increase greatly. Moreover, in order to decrease the dimension of problem, a new transformation function which transforms the decision space into a lower dimension search space (weight vector space) is designed. Furthermore, experiments are conducted on some benchmarks with 200, 500 and 1000 decision variables and the proposed algorithm SSVG is compared with three state-of-the-art algorithms: MOEA/DVA, WOF and LSMOF. The results show that the proposed algorithm outperforms the compared algorithms in term of convergence and diversity.**

***Index Terms*—large-scale multi-objective optimization, variable grouping, symmetric point, dimension reduce, problem transformation**

I. Introduction

Multi-objective optimization problems (MOPs) exist commonly in real world. With the increasing of the factors of the problems, the scale of the mathematical model constructed becomes larger and larger. These problems usually includes a large number of decision variables and are named large-scale multi-objective optimization problems. The features of these problems are that the decision space is very huge and the objectives conflict with each other. Generally speaking, large-scale MOPs are much more difficult to be solved than those with a few decision variables, since the search space exponentially increases with the increasing of the number of decision variables, which causes the so called the curse of dimensionality. Therefore, it is very challenging for multi-objective evolutionary algorithms to exploit the search space efficiently [1].

Up to date, many algorithms have been proposed to solve the large-scale multi-objective optimization problems. These algorithms can be classified into three categories as follows. The algorithms in the first category are based on decision variable grouping. They apply grouping mechanism to divide the decision variables into several groups, then optimize each group respectively. For instance, CCGDE3 [2] randomly divides the decision variables into some equal groups, and then uses GDE3 [3] to optimize each group. The algorithms in the second category are based on variable interaction analysis. The variable interaction analysis method divides the decision variables into different groups, and in each group we use a specific optimization strategy. For example, MOEA/DVA [4] divides decision variables into three groups, i.e., convergence-related, diversity-related and both convergence and diversity-related. And then they perform different optimization strategies for different groups. The algorithms in the third category are based on problem transformation. The large-scale multi-objective optimization problems are transformed into several small-scale multi-objective problems. WOF [5] and LSMOF [6] are two typical algorithms based on reformulation. For example, WOF divides the decision variables into some groups first, with every group affiliated with a weight variable. Thus the number of weight variables is smaller than that of original variables. And then WOF transforms optimizing the original variables into optimizing the weight variables. As a result, the transformed problem is smaller than the original one.

However, these algorithms have some drawbacks. The variable analysis-based methods consume a large amount of function evaluations due to the variable analysis. Without any prior knowledge, the grouping-based method may divide the decision variables into improper groups. Although the methods based on transformation decrease the number of dimensions and the difficulty of solving problems, the directions to search optimal solutions are not always effective. Furthermore, looking for some efficient search directions is difficult for large-scale multi-objective optimization problems. Motivated by this, we propose a method based on symmetric points search for solving large-scale multi-objective optimization problems.

The rest of this paper is organized as follows. In Section II, we introduce some related works and our motivation. The proposed SSVG is presented in Section III. We compare SSVG with state-of-the-art algorithms by experiments in Section IV. Finally, the conclusions are made in Section V.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

II. Related Works and Motivation

In this section, some basic concepts and definitions about large-scale multi-objective optimization are explained first. And then three related large-scale multi-objective optimization algorithms are briefly introduced. Finally, our motivation is presented.

A. Some Basic Concepts and Definitions

Definition 1. Multi-objective optimization problems: Many problems can be modeled as mathematical problems with some objectives that are always conflict. These problems with 2 or 3 objectives are often called multi-objective optimization problems (MOPs). They can be formulated as:

$$\begin{aligned} Z: \min F(x) = \left(f_1\left(x\right), f_2\left(x\right), \ldots, f_m\left(x\right)\right) \\ s.t. \quad x \in \Omega \subseteq R^n \end{aligned} \tag{1}$$

This kind of MOPs maps the decision space $\Omega = \{x \in R^n | g(x) \leq 0\}$ of dimension n to the objective space M of dimension m.

Definition 2. Large-scale Multi-objective optimization problem: If the dimension of the decision space is larger or equal to 100, an multi-objective optimization problem is called as a large-scale multi-objective optimization problem.

Since the objectives of MOPs are always conflict with each other, the algorithms for single objective optimization problems are no longer effective for multi-objective optimization problems. The aim of MOPs algorithms is looking for a Pareto set which makes objectives obtain optimal values as much as possible. The conflicts between objectives mean that interdependence exists among some decision variables. That is to say, if you change one decision variable, the optimal values of other decision variables will be changed at the same time. This increases the difficulty to solve this kind of problems.

B. MOEA/DVA

The MOEA/DVA [4] proposed in 2016 applies the interdependence analysis into optimization process, and the decision variables were divided into three groups: position variables, distance variables and mixed variables. As its name shown, the position variable group contains variables which contributes to convergence, and the distance variables play an important role in diversity. Decision variables which do not belong the above two groups are seen as the mixed variables.

Although the MOEA/DVA obtains the interdependence between decision variables and makes optimization more effective, the interdependence analysis consume a large amount of function evaluations. Relatively, the computation resource in optimization process may be restricted.

C. Weighted Optimization Framework(WOF)

The feature of large-scale multi-objective optimization is the high dimension of decision variables. In order to reduce the dimension of the decision space, the weighted optimization framework (WOF) [7] is first to divide the n dimensional decision variables $(x_1, x_2, \ldots, x_n)$ into γ sub-groups with $\gamma < n$. Each group will associate with a weight variable. And then the original problem with n dimensional decision space is transformed into a new problem of weight variables with γ dimensional weight space. Some metaheuristic optimizers were employed to optimize weight variables. According to transformation function, the weight variables can be affiliated with the original decision variables, changing the original solutions indirectly. As a consequence, the optimization of the weight variable in the same group can be regarded as the optimization of a subproblem in a subspace of the original decision space [7].

D. Large-scale Multi-objective Optimization Framework (LSMOF)

The large-scale multi-objective optimization framework (LSMOF) [6] was proposed in 2019 and was experimentally shown effective. The feature of LSMOF is using bi-directional search. It contains two steps, The first step is looking for a quash-optimal solution sets near the PS by using the problem transformation and the second step spreads the quash-optimal solution sets to approximate the true PF.

In the first step, LSMOF uses some optimal points as reference points, and each point is affiliated with two direction vectors (i.e. v_l and v_u) and two weight variables (i.e. λ_{11} and λ_{12}). Given two direction vectors and the reference points, we can find the optimal solutions along the direction vectors as follows:

$$\begin{aligned} p_1 = o + \lambda_{11} \frac{v_l}{\|v_l\|} l_{\max} \\ p_2 = t - \lambda_{12} \frac{v_u}{\|v_u\|} l_{\max} \end{aligned} \tag{2}$$

where o and t are the lower and upper bound points and $l_{max} = ||t - o||$.

Once the original problem is reformulated, LSMOF performs single-objective optimization of the weight variables in the reconstructed decision space and the reduced objective space [6]. The objective function is transformed into a new single-objective function:

$$\begin{aligned} z_{11}\left(\lambda_{11}\right) &= F\left(o + \lambda_{11} \frac{v_1}{\|v_l\|} l_{\max}\right) \\ z_{12}\left(\lambda_{12}\right) &= F\left(o + \lambda_{12} \frac{v_u}{\|v_u\|} l_{\max}\right) \end{aligned} \tag{3}$$

For more details, please refer to [6].

E. Motivation

Note that MOEA/DVA using the interdependence analysis consumes a great number of function evaluations before the start of the optimization. Because the decision space of large-scale multi-objective optimization problem is very huge, MOEA/DVA may be inefficient. WOF uses problem transformation mechanism to decrease the decision space, reducing difficulty of searching optimal solutions sets. However, there is only one search direction in the reduced subspaces, and some optimal solutions can not be found because these optimal solutions may not allocate in the associated subspaces.

Similarly, LSMOF only adopted the bi-direction vectors to guide the search, and it also enough.

Note that the distribution of the Pareto set in decision space may be uneven and unknown before the optimization. If the Pareto Front is complicated, the possibility to find the true Pareto set will decrease. It is crucial for large-scale multi-objective optimization problem algorithms to use multiple potential search directions. Motivated by this, this paper proposes a symmetric point search and variable grouping method for large-scale multi-objective optimization (SSVG).

III. The proposed method

In this section, we explain the proposed method in detail. We shall introduce a symmetric points search method and a new transformation function, which needs fewer parameters than those used in WOF.

A. The Symmetric Points Search and Variable Grouping Method (SSVG)

The aim of the symmetric points search method is increasing the search directions in the reduced subspaces. Similar to WOF, our method uses grouping mechanism to divide the high dimensional decision space into several low dimensional subspaces. An n dimensional vector $X = (x_1, x_2, \ldots, x_n)$ is divided into k sub-vectors $X = (X_1, X_2, \cdots, X_k)$, where $X_i \in R^{n_i}$ for $i = 1, 2, \cdots, k$. Each sub-vector X_i is assigned a weight variable w_i. For any sub-vector $X_i = (y_1, y_2, \cdots, y_{n_i})$, its n_i symmetric points with respect to each dimension are $Y_j = (y_1, y_2, \cdots, -y_j, \cdots, y_{n_i})$ for $j = 1, 2, \cdots, n_i$, respectively, and there are $\binom{n_i}{2}$ possible pairs of these n_i symmetric points. Randomly take m pairs from these symmetric points, where m is a predefined number. Each pair, say, the r pair of P and Q, defines one search direction by $d_{r+1} = P - Q$ for $r = 1, 2, \cdots, m$. Also, let $d_1 = X_i$. In this way, for each X_i, we can define $m + 1$ directions $d_1, d_2, \cdots, d_{m+1}$. In order to avoid defining too many search directions for each sub-vector, m should not be large. We take $m = 4$ in the experiments.

Fig. 1 gives an example to illustrate this process. Suppose the original point $X = (x_1, x_2, \ldots, x_{20})$ is divided into $X = (X_1, X_2, \cdots, X_{10})$, where $X_1 = (x_1, x_2)$. For X_1, $m = 1$ pair $Y_1 = (-x_1, x_2)$ and $Y_2 = (x_1, -x_2)$ of symmetric points is chosen. Two search directions $d_1 = (x_1, x_2)$ and $d_2 = Y_1 - Y_2$ can be defined and is shown in Figure 1. The search region around point X_1 is divided into four quadrants in the 2-D subspace. With changing weight w_1, the new solutions produced by the transformation function will change. By searching optimal weight w_1 along d_1, we can find the optimal solutions of the first and the third quadrants. But it is difficult to find the optimal solutions in the second and the forth quadrants only along the d_1. By using search direction d_2, we can search the optimal solutions in the second and the forth quadrants along the d_2. So searching the solutions along d_1 and d_2 will improve the diversity of solutions.

The SSVG contains two steps, where the first step aims to search the optimal solutions in the reduced decision space and the second step spreads the obtained solutions to maintain the diversity. The main framework is shown in Algorithm 1.

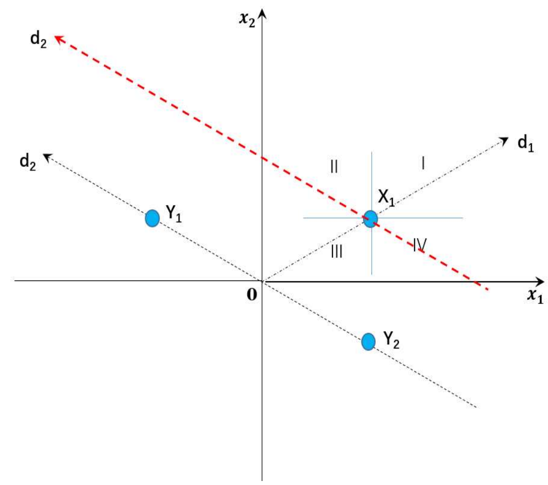

Fig. 1: An example of the symmetric points and search directions.

For SSVG, an original problem Z, a transform function ψ and a population size N are given. First, an initial population for original problem Z is produced. Then we select q different reference points with the largest crowding distance. In every iteration of the main loop (i.e.,lines 4-12 in algorithm 1), for every reference point, SMPSO [8] is applied to optimize the weight vector. The obtained optimal weight vectors are denoted by $wv_k(k = 1, 2, ..., q)$.

Next, the optimal weight vectors wv_k are combined with reference points to generate new solutions by a new transformation function, forming population P_{new1}. Then a number of directions are calculated by using symmetric points and we do the symmetric point search by SMPSO along these directions. The resulted population is denoted as P_{new2} (refer to algorithm 2 for detail). To keep the diversity of the population, we randomly select some original points from P_{new2} and combine the optimal weight vectors wv_k with the selected points to form a new population P_{new3}. Finally, the environment selection is used to select N (population size) best solutions as the final population.

B. A new transformation function

In this subsection, we propose a new transformation function for grouped variables. In [5], there are three transformation functions. The best function is the *p-Value* Transformation. But the expression of this function is a little complicated. The performance of the function is affected by parameter p greatly. A larger value of p may yield an invalid solution, and a smaller value of p may yield a solution which have no any improvement. To overcome this drawback, we introduce a new transformation function as follows:

$$x_{i,new} = \begin{cases} x_{i,old} - w_j \cdot \rho_{\min} & \text{if } \rho_{\min} \leq \rho_{\max} \\ x_{i,old} + w_j \cdot \rho_{\max} & \text{if } \rho_{\min} > \rho_{\max} \end{cases} \tag{4}$$

Algorithm 1 The main framework of SSVG

Input: Z: original problem ; ψ: transformation function; N: population size;
Output: optimal solutions set S

```
Initialization;
P ← initial population for original problem Z;
Q = {x_1, x_2, ..., x_q} ← selectReferencePoint (P, q); //select q different reference points;
repeat
for k = 1 to q do
    wv_k ← WeightvectorOptimization(P, Q, ψ); //use the Heuristic algorithm to optimize the weight vector
    P_new1 ← WeightTransformToDecision(wv_k, ψ, Q_k); //use the transformation function to apply the weight variable to decision vectors
    P_new2 ← SymmetricPointSearch(P_new1); // use Algorithm 2
    P_new3 ← ExtendPopulation(wv_k, Q_k); //keep the diversity of the population
    P_union ← P_new1 ⋃ P_new2 ⋃ P_new3;
end for
until σ * total Evaluations used;
S ← EnvironmentSelection(P_union, N);
return S;
```

Algorithm 2 SymmetricPointSearch(P)

Input: P: A population
Output: solutions P_{rs}

```
Initialization;
for i = 1 to r do
    r_2 ← calculate the symmetric points of the current population;
    d_2 ← calculate the intersecting direction using(4);
    P ← search the optimal results along the d_2;
    P_rs ← P_rs ⋃ P;
end for
return P_rs;
```

$$w_j \in [0,1]$$
$$\rho_{\min} = x_{i,old} - x_{i,\min}$$
$$\rho_{\max} = x_{i,\max} - x_{i,old}$$

where $\boldsymbol{x_{i,\min}}$and $\boldsymbol{x_{i,\max}}$ are the lower and upper bounds of $\boldsymbol{x_i}$ respectively. The w_j is the weight variable corresponding to group that the decision variable $\boldsymbol{x_i}$ belongs to. If $\boldsymbol{x_i}$ is closer to $\boldsymbol{x_{i,\min}}$, the new solution locates on the left of $\boldsymbol{x_i}$. Similarly, if $\boldsymbol{x_i}$ is closer to $\boldsymbol{x_{i,\max}}$, the new solution locates on the right of $\boldsymbol{x_i}$. Thus, the value of the new solution always locates in the interval [$\boldsymbol{x_{i,\min}}$,$\boldsymbol{x_{i,\max}}$].

IV. Experimental results

In this section, we conduct the experiments and compare the performance of the proposed algorithm SSVG with three state-of-the-art algorithms MOEA/DVA, WOF, and LSMOF on several widely used benchmarks ZDT1-ZDT4, ZDT6 [9] and UF1-10 [10]. We use PlatEMO [11] as the platform to investigate the performance of these algorithms.

The performance indicators used in this paper are the inverted generational distance (IGD) [12] and the hyper-volume indicator (HV) [13].

A. Experimental settings

For fairness, all algorithms to be compared are implemented in PlatEMO [11]. The population size of the algorithms is set to 100. The number of the FEs is set to 30000 for all compared algorithms. The parameters needed in MOEA/DVA, WOF, and LSMOF are set to defaults in their algorithms. Each test instance is run for 30 times independently.

B. Comparison and analysis of the results

In the experiments. the dimensions adopted are $n = 200$, $n = 500$, and $n = 1000$, respectively. The Wilcoxon rank sum test with a significance level of 0.05 is adopted to perform statistical analysis on test results, where the symbols "+", "-", and "=" indicate that the result of the compared algorithm is significantly better, significantly worse, and statistic similar to that of the proposed algorithm.

The statistics results of IGD values achieved by MOEA/DVA, WOFSMPSO, LSMOF and our method SSVG on ZDT problems are presented in Table I. It can be observed that most of the best performance results are obtained by SSVG. SSVG has achieved 9 out of 15 best results. LSMOF has achieved 3 out of 15 best results, and WOFSMPSO has achieved 3 best results. To be specific, SSVG achieves the best results on ZDT3, ZDT6, 200-dimension ZDT1, 500-dimension ZDT2, and 1000-dimension ZDT2. LSMOF achieves the best results on ZDT4. WOFSMPSO achieves the best results on 500-dimension ZDT1, 1000-dimension ZDT1 and 200-dimension ZDT2.

The statistics results of HV values achieved by MOEA/DVA, WOFSMPSO, LSMOF and our method SSVG on ZDT problems are listed in Table II. It can be seen that SSVG has achieved most of the best results. Specifically, SSVG obtains the best results on ZDT2, ZDT6, 200-dimension ZDT3, 500-dimension ZDT3, and 200-dimension ZDT1. LSMOF achieves the best results on ZDT4 and 1000-dimension ZDT3. WOFSMPSO achieves two best results on 500-dimension and 1000-dimension ZDT1.

Table III lists the IGD values obtained by the four above algorithms on UF1-10. SSVG has achieved 21 out of 30 best results. It performs best on UF1-3, 500-dimension UF5, 1000-dimension UF5, 200-dimension UF6, and UF7-9. WOFSMPSO outperforms others on UF4, 500-dimension UF6, and 1000-dimension UF6. It can be observed that the SSVG outperforms others on the 3-objective problems.

Table IV presents the HV values of the above algorithms on UF1-10. SSVG outperforms others on UF1-4, 500-dimension UF6, and UF7-9. The solutions obtained by SSVG are better than those of other three algorithms.

In summary, the proposed SSVG is effective and shows a competitive performance in comparison with MOEA/DVA, WOFSMPSO and LSMOF.

TABLE I:
IGD VALUE OF MOEA/DVA, LSMOF, WOFSMPSO, AND SSVG ON TWO-OBJECTIVE ZDT1-4 AND ZDT6, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS SHOWN IN BOLD FONT

Problem	M	D	MOEA/DVA	LSMOF	WOFSMPSO	SSVG
ZDT1	2	200	6.4244e+2(1.61e+1)-	2.4300e-1(1.21e-1)-	1.9656e-2(8.08e-2)=	**4.8079e-3(2.26e-4)**
	2	500	1.9667e+3(2.58e+1)-	3.0307e-1(1.17e-1)-	**3.2719e-2(1.53e-1)=**	1.1630e-1(2.89e-1)
	2	1000	4.1841e+3(3.92e+1)-	4.1324e-1(1.44e-1)-	**1.4419e-1(3.17e-1)=**	2.8083e-1(3.94e-1)
ZDT2	2	200	6.6528e+2(1.46e+1)-	2.4982e-1(1.06e-1)-	**4.9395e-3(3.24e-4)=**	4.9741e-3(2.21e-4)
	2	500	2.0060e+3(1.91e+1)-	4.1752e-1(8.95e-2)-	1.1620e-1(2.32e-1)=	**8.5777e-2(2.09e-1)**
	2	1000	4.2370e+3(3.40e+1)-	4.7065e-1(9.42e-2)-	3.2739e-1(3.07e-1)=	**2.2272e-1(2.92e-1)**
ZDT3	2	200	6.5207e+2(1.21e+1)-	3.8837e-1(1.49e-1)-	2.0173e-1(2.50e-1)=	**1.2333e-1(2.15e-1)**
	2	500	1.9626e+3(2.66e+1)-	4.6259e-1(1.16e-1)-	3.8209e-1(2.72e-1)=	**2.3355e-1(2.90e-1)**
	2	1000	4.1978e+3 (3.52e+1)-	4.9402e-1(1.39e-1)=	4.9054e-1(2.80e-1)=	**4.2915e-1(3.28e-1)**
ZDT4	2	200	2.7335e+3(5.30e+1)-	**3.1120e-2(2.59e-2)+**	5.2209e+2(1.01e+2)-	4.3475e+2(1.18e+2)
	2	500	8.1138e+3(7.18e+1)-	**3.8589e-2(2.55e-2)+**	2.3982e+3(2.53e+2)-	1.9614e+3(4.19e+2)
	2	1000	1.7255e+4(1.73e+2)-	**4.8516e-2(2.19e-2)+**	6.0303e+3(5.44e+2)-	5.5892e+3(7.17e+2)
ZDT6	2	200	2.6872e+1(1.29e-1)-	4.5647e-3(3.49e-4)-	3.8812e-3(1.92e-4)=	**3.8481e-3(1.89e-4)**
	2	500	3.5169e+1(1.08e-1)-	4.7883e-3(1.98e-3)-	3.9133e-3(2.37e-4)=	**3.8797e-3 (2.06e-4)**
	2	1000	4.2365e+1 (8.89e-2) -	7.4600e-3 (7.98e-3) -	3.9009e-3 (1.26e-4) =	**3.8694e-3 (1.63e-4)**
$+/-/=$			0/15/0	3/11/1	0/3/12	

TABLE II:
HV VALUE OF MOEA/DVA, LSMOF, WOFSMPSO, AND SSVG ON TWO-OBJECTIVE ZDT1-4 AND ZDT6, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS SHOWN IN BOLD FONT

Problem	M	D	MOEA/DVA	LSMOF	WOFSMPSO	SSVG
ZDT1	2	200	0.0000e+0(0.00e+0)-	5.7555e-1(7.43e-2)-	7.1007e-1(5.05e-2)=	**7.1938e-1(2.43e-4)**
	2	500	0.0000e+0(0.00e+0)-	5.3786e-1(7.59e-2)-	**6.9840e-1(1.15e-1)=**	6.3548e-1(2.17e-1)
	2	1000	0.0000e+0(0.00e+0)-	4.6068e-1(1.05e-1)-	**6.1456e-1(2.38e-1)=**	5.1247e-1(2.96e-1)
ZDT2	2	200	0.0000e+0(0.00e+0)-	2.2993e-1(7.78e-2)-	4.4393e-1(2.91e-4)=	**4.4402e-1(2.10e-4)**
	2	500	0.0000e+0(0.00e+0)-	1.3798e-1(4.18e-2)-	3.7632e-1(1.39e-1)=	**3.9654e-1(1.22e-1)**
	2	1000	0.0000e+0(0.00e+0)-	1.2073e-1(3.50e-2)-	2.5567e-1(1.79e-1)=	**3.1504e-1(1.72e-1)**
ZDT3	2	200	0.0000e+0(0.00e+0)-	5.2856e-1(1.60e-1)=	5.1387e-1(1.04e-1)=	**5.4419e-1 (8.69e-2)**
	2	500	0.0000e+0(0.00e+0)-	4.4979e-1(1.31e-1)-	4.3273e-1(1.59e-1)=	**4.8784e-1 (1.47e-1)**
	2	1000	0.0000e+0(0.00e+0)-	**4.1721e-1(1.50e-1)=**	3.6844e-1(2.08e-1)=	3.7197e-1(1.93e-1)
ZDT4	2	200	0.0000e+0(0.00e+0)-	**7.0026e-1(1.56e-2)+**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
	2	500	0.0000e+0(0.00e+0)-	**6.9490e-1(1.62e-2)+**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
	2	1000	0.0000e+0(0.00e+0)-	**6.8851e-1(1.37e-2)+**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
ZDT6	2	200	0.0000e+0(0.00e+0)-	3.8742e-1(3.31e-4)-	3.8809e-1(1.93e-4)=	**3.8813e-1(2.13e-4)**
	2	500	0.0000e+0(0.00e+0)-	3.8711e-1(1.96e-3)-	3.8801e-1(2.65e-4)=	**3.8809e-1(2.15e-4)**
	2	1000	0.0000e+0(0.00e+0)-	3.8432e-1(8.26e-3)-	3.8804e-1(1.25e-4)=	**3.8809e-1(1.72e-4)**
$+/-/=$			0/12/3	3/10/2	0/0/15	

TABLE III:
IGD VALUE OF MOEA/DVA, LSMOF, WOFSMPSO, AND SSVG ON TWO-OBJECTIVE UF1-7 AND THREE-OBJECTIVE UF8-10, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS SHOWN IN BOLD FONT

Problem	M	D	MOEA/DVA	LSMOF	WOFSMPSO	SSVG
UF1	2	200	1.4356e+0(6.06e-2)-	2.0676e-1(1.78e-2)-	1.1033e-1(7.84e-3)-	**1.0108e-1(5.81e-3)**
	2	500	1.9342e+0(4.20e-2)-	2.6937e-1(2.11e-2)-	1.1027e-1(1.07e-2)-	**1.0237e-1(8.46e-3)**
	2	1000	2.1306e+0(3.49e-2)-	2.8972e-1(1.82e-2)-	1.1123e-1(8.86e-3)-	**1.0224e-1(8.29e-3)**
UF2	2	200	6.6944e-1(2.34e-2)-	9.9280e-2(2.42e-2)-	8.4781e-2(4.77e-3)=	**8.2428e-2(3.34e-3)**
	2	500	8.5969e-1(1.42e-2)-	1.1729e-1(3.24e-2)-	8.4877e-2(3.68e-3)=	**8.4259e-2(4.72e-3)**
	2	1000	9.3068e-1(8.77e-3)-	1.2465e-1(3.47e-2)-	8.3327e-2(3.06e-3)=	**8.2699e-2(4.22e-3)**
UF3	2	200	8.4520e-1(2.67e-2)-	1.6467e-1(2.13e-3)-	7.2305e-2(4.58e-3)-	**6.0904e-2(2.18e-3)**
	2	500	1.0347e+0(2.25e-2)-	1.4117e-1(1.57e-3)-	4.0928e-2(3.77e-3)-	**3.3156e-2(2.07e-3)**
	2	1000	1.1050e+0(1.64e-2)-	1.3340e-1(2.22e-3)-	3.0033e-2(3.33e-3)-	**2.3620e-2(2.22e-3)**
UF4	2	200	1.8706e-1(1.39e-3)-	7.1716e-2(1.93e-3)+	**6.4733e-2(6.52e-3)+**	1.4908e-1(7.19e-2)
	2	500	2.1288e-1(9.24e-4)-	7.2861e-2(9.96e-4)+	**6.3905e-2(7.71e-3)+**	1.4152e-1(7.41e-2)
	2	1000	2.2146e-1(6.50e-4)-	7.3016e-2(1.05e-3)+	**6.5655e-2(6.05e-3)+**	1.3994e-1(7.56e-2)
UF5	2	200	5.5145e+0(1.36e-1)-	**1.5898e+0(3.27e-1)=**	1.7773e+0(2.24e-1)-	1.6608e+0(2.03e-1)
	2	500	6.7153e+0(1.08e-1)-	2.3079e+0(2.88e-1)-	1.9926e+0(9.72e-2)=	**1.9616e+0(8.95e-2)**
	2	1000	7.1427e+0(8.46e-2)-	2.7000e+0(1.49e-1)-	2.0976e+0(6.45e-2)=	**2.0686e+0(6.81e-2)**
UF6	2	200	5.9192e+0(3.22e-1)-	1.0095e+0(1.74e-1)-	3.6630e-1(6.62e-2)=	**3.4872e-1(7.32e-2)**
	2	500	7.8126e+0(1.51e-1)-	1.2801e+0(2.23e-1)-	**3.6235e-1(6.39e-2)=**	3.6967e-1(8.05e-2)
	2	1000	8.5818e+0(1.40e-1)-	1.4150e+0(1.77e-1)-	**3.4259e-1(2.79e-2)=**	3.7085e-1(7.56e-2)
UF7	2	200	1.5235e+0(5.76e-2)-	4.2204e-1(9.22e-3)-	1.1085e-1(7.26e-2)-	**8.9352e-2(4.14e-2)**
	2	500	1.9847e+0(3.03e-2)-	4.6398e-1(9.34e-3)-	1.0219e-1(5.40e-2)-	**8.6153e-2(3.62e-3)**
	2	1000	2.1730e+0(3.00e-2)-	4.8163e-1(7.92e-3)-	8.8844e-2(4.34e-3)-	**8.6107e-2(3.54e-3)**
UF8	3	200	3.0266e+0(1.26e-1)-	5.5835e-1(1.57e-2)-	3.2193e-1(2.74e-2)-	**2.7440e-1(2.42e-2)**
	3	500	3.9306e+0(9.77e-2)-	6.1040e-1(2.41e-2)-	3.2216e-1(2.90e-2)-	**2.6367e-1(1.94e-2)**
	3	1000	4.3216e+0(9.05e-2)-	6.3936e-1(2.29e-2)-	3.1574e-1(2.77e-2)-	**2.5773e-1(1.84e-2)**
UF9	3	200	3.2020e+0(1.21e-1)-	5.5111e-1(8.38e-3)-	5.5609e-1(2.14e-2)-	**5.0721e-1(5.40e-2)**
	3	500	4.0787e+0(9.81e-2)-	5.9401e-1(1.32e-2)-	5.6470e-1(2.55e-2)-	**4.5730e-1(7.97e-2)**
	3	1000	4.4472e+0(6.29e-2)-	6.2133e-1(1.75e-2)-	5.6740e-1(2.19e-2)-	**4.5043e-1(6.02e-2)**
UF10	3	200	1.5720e+1(5.92e-1)-	**1.6644e+0(2.88e-1)+**	3.5834e+0(2.52e-1)=	3.6735e+0(2.51e-1)
	3	500	1.9465e+1(4.61e-1)-	**2.7168e+0(3.57e-1)+**	3.9659e+0(2.81e-1)=	3.9153e+0(2.03e-1)
	3	1000	2.0797e+1(3.24e-1)-	**3.3647e+0(3.20e-1)+**	4.1293e+0(2.68e-1)=	4.0257e+0(2.57e-1)
$+/-/=$			0/30/0	6/23/1	3/16/11	

V. Conclusion

This paper proposes a new method based on the symmetric points search and variable grouping for large-scale multi-objective optimization problems, named SSVG. The proposed SSVG adopts two main steps, where in the first step we use grouping mechanism to divide the decision space into several sub-space and reformulate the problem into a lower dimensional problem. In each subspace, a weight variable is assigned to this subspace, and the symmetric points of the original points form a number of search directions. An embedded MOEA searches the solutions along the obtained search directions. In the second step, the solutions obtained by the first step are extended to whole space solutions and the diversity of solutions are kept. A number of experiments are conducted on widely used benchmarks, and the proposed SSVG is compared with three state-of-the-art large-scale MOEAs, namely, MOEA/DVA, WOF, and LSMOF. The statistical results show that SSVG performs better than the compared algorithms and has good potential in solving large-scale multi-objective optimization problems.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NO.61872281) and Key Natural Science Foundation of Shaanxi Province (No. 2016JZ022) and the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University(NO.CS2020-08).

TABLE IV:
HV VALUE OF MOEA/DVA, LSMOF, WOFSMPSO, AND SSVG ON TWO-OBJECTIVE UF1-7 AND THREE-OBJECTIVE UF8-10, WHERE THE BEST RESULT ON EACH TEST INSTANCE IS SHOWN IN BOLD FONT

Problem	M	D	MOEA/DVA	LSMOF	WOFSMPSO	SSVG
UF1	2	200	0.0000e+0(0.00e+0)-	4.0158e-1(2.64e-2)-	5.5375e-1(1.52e-2)-	**5.7724e-1(1.10e-2)**
	2	500	0.0000e+0(0.00e+0)-	3.3122e-1(2.80e-2)-	5.5489e-1(2.07e-2)-	**5.7424e-1(1.62e-2)**
	2	1000	0.0000e+0(0.00e+0)-	3.1587e-1(2.60e-2)-	5.5196e-1(1.60e-2)-	**5.7492e-1(1.42e-2)**
UF2	2	200	7.2349e-2(8.50e-3)-	5.9410e-1(4.59e-2)-	6.2384e-1(6.07e-3)=	**6.2681e-1(4.21e-3)**
	2	500	1.1589e-2(3.83e-3)-	5.6538e-1(6.21e-2)-	6.2438e-1(4.60e-3)=	**6.2487e-1(5.98e-3)**
	2	1000	1.8613e-3(1.33e-3)-	5.5275e-1(6.66e-2)-	6.2620e-1(4.05e-3)=	**6.2693e-1(5.50e-3)**
UF3	2	200	5.5263e-3(3.65e-3)-	5.2814e-1(2.45e-3)-	6.3470e-1(5.21e-3)-	**6.4785e-1(2.37e-3)**
	2	500	0.0000e+0(0.00e+0)-	5.5792e-1(2.12e-3)-	6.7227e-1(4.31e-3)-	**6.8126e-1(2.38e-3)**
	2	1000	0.0000e+0(0.00e+0)-	5.6760e-1(2.56e-3)-	6.8550e-1(3.89e-3)-	**6.9304e-1(2.55e-3)**
UF4	2	200	2.0648e-1(1.05e-3)-	3.5244e-1(1.68e-3)=	3.5848e-1(7.47e-3)-	**3.7581e-1(5.98e-2))**
	2	500	1.8430e-1(6.47e-4)-	3.4979e-1(1.72e-3)=	3.5977e-1(8.39e-3)-	**3.7090e-1(6.01e-2)**
	2	1000	1.7763e-1(3.90e-4)-	3.4911e-1(1.79e-3)-	3.5660e-1(7.63e-3)-	**3.6811e-1(5.85e-2)**
UF5	2	200	**0.0000e+0(0.00e+0)=**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
	2	500	**0.0000e+0(0.00e+0)=**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
	2	1000	**0.0000e+0(0.00e+0)=**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
UF6	2	200	0.0000e+0(0.00e+0)=	1.2520e-3(5.29e-3)-	**1.6310e-1(2.28e-2)=**	1.6301e-1(2.47e-2)
	2	500	0.0000e+0(0.00e+0)=	1.4413e-3(5.51e-3)-	1.4700e-1(2.73e-2)=	**1.5238e-1(2.75e-2)**
	2	1000	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	**1.6075e-1(2.30e-2)+**	1.4520e-1 (2.71e-2)
UF7	2	200	0.0000e+0(0.00e+0)=	1.7838e-1(1.03e-2)-	4.4310e-1(5.84e-2)-	**4.6395e-1(3.26e-2)**
	2	500	0.0000e+0(0.00e+0)=	1.3328e-1(7.27e-3)-	4.4994e-1(4.28e-2)-	**4.6485e-1(6.64e-3)**
	2	1000	0.0000e+0(0.00e+0)=	1.1884e-1(5.16e-3)-	4.6016e-1(7.46e-3)-	**4.6550e-1(5.59e-3)**
UF8	3	200	0.0000e+0(0.00e+0)=	1.0334e-1(8.57e-3)-	1.8828e-1(2.35e-2)-	**2.2589e-1(2.50e-2)**
	3	500	0.0000e+0(0.00e+0)=	7.6092e-2(1.41e-2)-	1.8970e-1(2.94e-2)-	**2.3907e-1(2.78e-2)**
	3	1000	0.0000e+0(0.00e+0)=	6.1445e-2(1.13e-2)-	1.9534e-1(2.36e-2)-	**2.4292e-1(2.78e-2)**
UF9	3	200	0.0000e+0(0.00e+0)=	2.1172e-1(8.64e-3)-	2.4915e-1(2.66e-2)-	**3.0589e-1(6.62e-2)**
	3	500	0.0000e+0(0.00e+0)=	1.7193e-1(1.15e-2)-	2.4269e-1(3.00e-2)-	**3.5169e-1(6.45e-2)**
	3	1000	0.0000e+0(0.00e+0)=	1.5072e-1(1.39e-2)-	2.4121e-1(2.63e-2)-	**3.7074e-1(6.09e-2)**
UF10	3	200	**0.0000e+0(0.00e+0)=**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
	3	500	**0.0000e+0(0.00e+0)=**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
	3	1000	**0.0000e+0(0.00e+0)=**	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=	0.0000e+0(0.00e+0)=
$+/-/=$			0/24/6	0/22/8	1/18/11	

References

[1] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient largescale multi-objective optimization based on a competitive swarm optimizer," *IEEE Transactions on Cybernetics*, pp. 1–13, 2019.

[2] L. M. Antonio and C. A. C. Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 2758–2765.

[3] S. Kukkonen and J. Lampinen, "Gde3: The third evolution step of generalized differential evolution," in *2005 IEEE congress on evolutionary computation*, vol. 1. IEEE, 2005, pp. 443–450.

[4] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 275–298, 2015.

[5] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 260–275, 2018.

[6] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, and X. Yao, "Accelerating large-scale multi-objective optimization via problem reformulation," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[7] R. Cheng, Y. Jin, M. Olhofer *et al.*, "Test problems for large-scale multiobjective and many-objective optimization." *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4108–4121, 2017.

[8] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. C. Coello, and E. Alba, "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *Computational intelligence in miulti-criteria decision-making, 2009. mcdm '09. ieee symposium on*, 2009.

[9] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[10] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the cec 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang*

technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report, vol. 264, 2008.

[11] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[12] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 892–899.

[13] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE transactions on evolutionary computation*, vol. 10, no. 1, pp. 29–38, 2006.