A Multiobjective Evolutionary Algorithm Based on Decision Variable Analyses for Multiobjective Optimization Problems With Large-Scale Variables

Xiaoliang Ma, Fang Liu, *Senior Member, IEEE*, Yutao Qi, Xiaodong Wang, Lingling Li, Licheng Jiao, *Senior Member, IEEE*, Minglei Yin, and Maoguo Gong, *Member, IEEE*

***Abstract*—State-of-the-art multiobjective evolutionary algorithms (MOEAs) treat all the decision variables as a whole to optimize performance. Inspired by the cooperative coevolution and linkage learning methods in the field of single objective optimization, it is interesting to decompose a difficult high-dimensional problem into a set of simpler and low-dimensional subproblems that are easier to solve. However, with no prior knowledge about the objective function, it is not clear how to decompose the objective function. Moreover, it is difficult to use such a decomposition method to solve multiobjective optimization problems (MOPs) because their objective functions are commonly conflicting with one another. That is to say, changing decision variables will generate incomparable solutions. This paper introduces interdependence variable analysis and control variable analysis to deal with the above two difficulties. Thereby, an MOEA based on decision variable analyses (DVAs) is proposed in this paper. Control variable analysis is used to recognize the conflicts among objective functions. More specifically, which variables affect the diversity of generated solutions and which variables play an important role in the convergence of population. Based on learned variable linkages, interdependence variable analysis decomposes decision variables into a set of low-dimensional subcomponents. The empirical studies show that DVA can improve the solution quality on most difficult MOPs. The code and supplementary material of the proposed algorithm are available at http://web.xidian.edu.cn/fliu/paper.html.**

***Index Terms*—Cooperative coevolution, decision variable analysis (DVA), interacting variables, multiobjective optimization, problem decomposition.**

Manuscript received March 29, 2014; revised August 29, 2014, December 13, 2014, and April 4, 2015; accepted July 6, 2015. Date of publication July 13, 2015; date of current version March 29, 2016. This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2013CB329402; in part by the National Natural Science Foundation of China under Grants 61173090, 61173092, 61303119, 61271302, 61272282, 61273317, 61271301, 61272279, 61001202, 61072106, 61072139, 61203303, and 61003199; in part by the National Research Foundation for the Doctoral Program of Higher Education of China under Grants 20100203120008 and 20110203110006; in part by the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) under Grant B07048; in part by the Fundamental Research Funds for the Central Universities under Grants K5051203002 and K5051203007; and in part by the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170.

X. Ma, F. Liu, Y. Qi, X. Wang, and M. Yin are with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China, also with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China, and also with the International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China (e-mail: maxiaoliang@yeah.net; f63liu@163.com; qi_yutao@163.com; 570629903@qq.com; axhiao@gmail.com).

L. Li, L. Jiao, and M. Gong are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China, and also with the International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China (e-mail: linglingxidian@gmail.com; lchjiao@mail.xidian.edu.cn; gong@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2015.2455812

I. Introduction

HAVING the advantage of generating a number of representatively approximate solutions in a single run, evolutionary algorithms (EAs) have been widely used for multiobjective optimization problems (MOPs) [1]. Currently, state-of-the-art multiobjective EAs (MOEAs) [2]–[4] pay more attention to keeping the diversity of obtained solutions in the objective space and treat all the decision variables as a whole to optimize. Due to the complexity and difficulty of MOP, it is interesting to investigate the ways of simplifying a given difficult MOP. A major factor contributing to the complexity and difficulty of an optimization problem is the number of decision variables [5]. Inspired by the cooperative coevolution [6]–[9] and linkage learning methods [10], [11], a desirable way is decomposing each objective function of MOP with high-dimensional variables into a number of simpler and low-dimensional subfunctions. If such a decomposition exists, optimizing the original function is equal to solving each subfunction separately. A major difficulty in the above "divide-and-conquer" strategy is how to select a good decomposition to keep the interdependencies among different subfunctions minimal. Although decomposition has an important effect on the performance of cooperative coevolution and linkage learning algorithms, there is usually not enough knowledge about the hidden structure of a given problem to help the algorithm designer discover a suitable decomposition. Therefore, it is necessary to devise an algorithm which can detect the interaction among decision variables in order to divide the decision variables. Interdependence variable analysis in this paper is developed for this purpose.

It is not trivial to generalize such divide-and-conquer strategy proposed in single objective optimization problem (SOP) to solve MOP because the objective functions of an MOP are conflicting with one another. The conflicts among the objective functions here refer to incomparable solutions to be generated by changing decision variables. The conflicts mean that the aim of MOP is to find a set of Pareto-optimal solutions rather than a single optimal solution as in SOP. Due to position variables and mixed variables [12] having an effect on the spread of the generated solutions, both kinds of variables are used as the root of conflicts among the objective functions.

Based on the above control analysis of decision variables and interdependent analysis between two variables, we propose an MOEA based on decision variable analyses (MOEA/DVAs). Based on control analysis of decision variables, MOEA/DVA decomposes a complicated MOP into a set of simpler sub-MOPs. Based on interdependent analysis between two variables, decision variables are decomposed into several low-dimensional subcomponents. Each sub-MOP independently optimizes subcomponents one by one. Therefore, MOEA/DVA is expected to have an advantage over most MOEAs which optimize all of the decision variables as a whole.

The major contributions of this paper are listed as follows.

1) In order to help the reader to understand the concept of variable interdependence, two necessary conditions for it are provided.
2) In order to learn the conflicts among objective functions, the concepts of position variable and mixed variable [12] are used. Additionally, based on position variables and mixed variables rather than weight vectors, this paper offers a new decomposition scheme to convert a difficult MOP into a set of simpler sub-MOPs.
3) With a good theoretical basis on interacting variables, this paper tries to decompose the difficult MOPs with high-dimensional variables into a set of simpler sub-MOPs with low-dimensional subcomponents.
4) This paper proves that the objective functions of continuous ZDT and DTLZ problems are separable functions. The interactions between two decision variables are sparse and focus on the mixed variable(s) for unconstrained multiobjective objective functions (UF) problems [13] of IEEE Congress on Evolutionary Computation (CEC) 2009 competition.

The rest of this paper is organized as follows. Section II introduces several related backgrounds about the definitions and notations of multiobjective optimization, variable linkage, control property of decision variable, linkage learning techniques for decision variable, and dividing techniques of variable based on the learning linkages. Section III describes DVAs and the proposed algorithm MOEA/DVA. Section IV illustrates and analyzes the experimental results. Section V concludes this paper.

II. Related Work

This section introduces two aspects of related research backgrounds. One is SOP. The related research backgrounds of SOP include separability and nonseparability of decision variables, various linkage learning methods, and different dividing methods for decision variables based on the learning linkages. The other is MOP. The related research backgrounds of MOP include the definitions and notations of MOP, regularity property [14] of continuous MOPs, and the control property of decision variables.

A. Separability and Nonseparability of Decision Variables

Definition 1: $f(\mathbf{x})$ is called a separable function [5] if and only if each decision variable $x_i, i = 1, \ldots, n$ can be optimized independently

$$\arg\min_{(x_1,\ldots,x_n)} f(x_1,\ldots,x_n) = \Big[\arg\min_{x_1} f(x_1,\ldots,x_i,\ldots,x_n), \ldots, \arg\min_{x_i} f(x_1,\ldots,x_i,\ldots,x_n), \ldots, \arg\min_{x_n} f(x_1,\ldots,x_i,\ldots,x_n)\Big]. \quad (1)$$

Otherwise, $f(\mathbf{x})$ is called a nonseparable function.

Function $f(x_1, x_2) = x_1 + x_2, x_1, x_2 \in [0, 1]$ is taken to explain the symbols in Definition 1. $\arg\min_{(x_1,x_2)} f(x_1, x_2)$ represents the optimal solution of $f(x_1, x_2)$ in decision space. It is easy to calculate $\arg\min_{(x_1,x_2)} f(x_1, x_2) = [0, 0]$. When x_2 is fixed, $\arg\min_{x_1} f(x_1, x_2)$ indicates the optimal solution of $f(x_1, x_2)$ on x_1. It is easy to calculate $\arg\min_{x_1} f(x_1, x_2) = 0$ for arbitrary $x_2 \in [0, 1]$ and $\arg\min_{x_2} f(x_1, x_2) = 0$ for arbitrary $x_1 \in [0, 1]$. Therefore, for the function $f(x_1, x_2) = x_1 + x_2, x_1, x_2 \in [0, 1]$, we have $\arg\min_{(x_1,x_2)} f(x_1, x_2) = [0, 0] = [\arg\min_{x_1} f(x_1, x_2), \arg\min_{x_2} f(x_1, x_2)]$. So, $f(x_1, x_2) = x_1 + x_2, x_1, x_2 \in [0, 1]$ is a separable function.

Definition 1 means that a separable function can be solved by optimizing variables one by one. Separability means that each variable is independent of any other variable. Other definitions on separable function and nonseparable function can be found in [7] and [12]. The sphere function, generalized Rastrigin's function, generalized Griewank's function, and Ackley's function [15], [16] are the representatives of separable functions. Basically, separability function means that the decision variables involved in the problem can be optimized independent of any other variable, while nonseparability function means that there exist interactions between at least two decision variables.

Variable dependencies are an important aspect of a problem and they describe the structure of a problem. If the variable dependencies of a problem are known in advance, it is easy to divide the decision variables into several subcomponents. Therefore, it is beneficial to solve a difficult problem with high-dimensional variables by optimizing several simpler subproblems with low-dimensional subcomponents separately. However, the variable dependencies of a problem are often unknown in advance. Moreover, the definition of "interdependent variables" is not unique. Yu *et al.* [17] offered

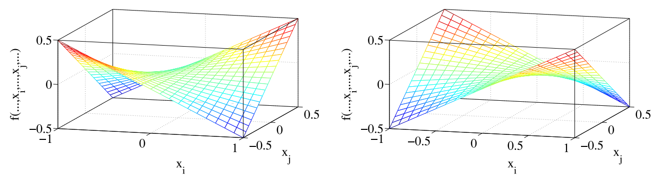


Fig. 1. Plot of two instances that exist interacting decision variables x_i and x_j.

that two decision variables are interacted if and only if the associated subproblem cannot be optimized without the information carried by both decision variables. Weise *et al.* [5] suggested that two decision variables interact with each other if the effect of varying one decision variable on the fitness relies on the value of the other decision variable. Different from the two above qualitative definitions, the following quantitative definition of interdependent variables is used in this paper.

Definition 2: Two decision variables x_i and x_j are interacting [18] if there exist $\mathbf{x}$, a_1, a_2, b_1, and b_2 meeting

$$f(\mathbf{x})|_{x_i=a_2,x_j=b_1} < f(\mathbf{x})|_{x_i=a_1,x_j=b_1} \wedge$$
$$f(\mathbf{x})|_{x_i=a_2,x_j=b_2} > f(\mathbf{x})|_{x_i=a_1,x_j=b_2} \tag{2}$$

where $f(\mathbf{x})|_{x_i=a_2,x_j=b_1} \triangleq f(x_1,\ldots,x_{i-1}, a_2,\ldots,x_{j-1}, b_1,\ldots,x_n)$.

Definition 2 can be derivable from the definition of nonseparability function suggested by Yang *et al.* [7]. Among these different definitions of "interdependent variables," we select Definition 2 as the definition of interdependent variables as this definition is quantitative and easy to use.

In other words, if there exist $\mathbf{x}$, a_1, a_2, b_1, and b_2 such that a strong dominance relationship between $[f(\mathbf{x})|_{x_i=a_2,x_j=b_1}, f(\mathbf{x})|_{x_i=a_1,x_j=b_2}]$ and $[f(\mathbf{x})|_{x_i=a_1,x_j=b_1}, f(\mathbf{x})|_{x_i=a_2,x_j=b_2}]$ is established, then two decision variables x_i and x_j are interacting. The definition of dominance can be found in Section II-D. Fig. 1 illustrates an example where two decision variables x_i and x_j are interacting. When $a_1 = -1$, $a_2 = 1$, $b_1 = -0.5$, and $b_2 = 0.5$, the formula 2 is established. Definition 2 is used to judge the interaction relationship between two decision vectors in our proposed algorithm.

A nonseparable function $f(\mathbf{x})$ is called fully nonseparable if any two different decision variables x_i and x_j are interacting. Schwefel's function 2.22, generalized Griewank function, and Ackley's function [15], [16] are the fully nonseparable functions. Between separable and fully nonseparable functions, there exist various partially separable functions [16], [19]. A function $f : \mathbf{R}^n \to \mathbf{R}$ is k-nonseparable if at most k of its decision variables $\mathbf{x}$ are not independent. In general, the bigger the degree of nonseparability, the harder a function can be solved [5]. Real-world optimization problems will be most likely made up of several independent modules [16], [19]. These problems are known as partially separable functions. An interesting point for this kind of problems is that a difficult function with high-dimensional decision variables can be decomposed into several simple subfunctions with low-dimensional subcomponents. Therefore, partially separable functions have attracted much attention in the fields of optimization [20] and evolutionary computation [16], [21].

B. *Interdependence Detecting Techniques for Decision Variables*

For problems with modular feature, if the algorithm can learn the problem structure and decompose the function accordingly, the difficulty to solve the problem will be reduced rapidly [22]. Thus, the key issue of reducing the difficulty of a problem is to detect the variable interactions. According to the suggestion of Yu *et al.* [17] and Omidvar *et al.* [21], linkage detecting techniques are divided into four major categories: 1) perturbation; 2) interaction adaptation; 3) model building; and 4) random.

1) Perturbation: These approaches detect interaction by perturbing decision variables and study the change of fitness due to such perturbations. The typical perturbation methods include the following two steps. The first step is to perturb decision variables and detect interactions among decision variables. The second step is to combine the decision variables with high interdependence within the same subcomponent to optimize. Examples of this kind of methods include linkage identification by nonlinear check (LINC) [23], linkage identification by nonlinear check for real-valued genetic algorithms (LINC-R) [24], adaptive coevolutionary optimization [25], and cooperative coevolution with variable interaction learning [18]. Our proposed interdependence variable analysis can be considered as a perturbation method.

2) Interaction Adaptation: These methods incorporate the interdependence detecting technique into the individual encoding and solve the problem simultaneously. Individuals with a tighter grouping of interdependent variables are assigned higher reproduction probability. Typical examples include the linkage learning genetic algorithm [26] and linkage evolving genetic operator [27].

3) Model Building: The classical framework of model building methods includes five steps: a) initializing an evolutionary population randomly; b) choosing a number of promising solutions; c) model building based on those selected promising solutions; d) sample new trial solutions from the learning model; and e) repeating steps b)–e) until the stopping criterion is met. Typical representatives include estimation of distribution algorithms (EDAs) [28], compact genetic algorithms [29], Bayesian optimization algorithms (BOAs) [30], and dependency structure matrix (DSM)-driven genetic algorithms [17].

4) Random Methods: Different from the above three methods, these approaches do not use intelligent procedure to detect the interactions among decision variables [7], [21]. They randomly permute the variables to improve the probability of putting the interacting variables into the same subcomponent [7].

C. *Dividing Techniques Based on the Learning Linkages*

In this section, we introduce two dividing methods for decision variables. The first one is dividing decision variables with interaction into the same subcomponent [21]. This dividing technique is effective for the problem with modularity. However, this dividing technique may not be the best for the problems with overlap and hierarchy structure [17].

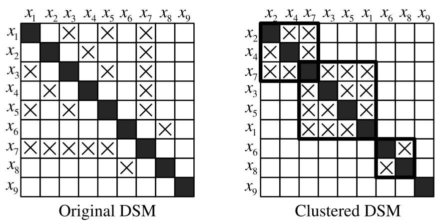


Fig. 2. Example of DSM clustering.

The second one is dividing variables based on DSM clustering technique [17]. A DSM is a matrix constructed by the interaction between two decision variables. DSM clustering technique is popular in architectural improvement of product design and development. The objective of DSM clustering is to obtain a clustering arrangement keeping maximal interaction within the same cluster but minimal interaction between clusters. Fig. 2 gives an example for DSM clustering.

In order to make a tradeoff between the accuracy and complexity of the clustering arrangement, Yu *et al.* [31] proposed a metric based on the minimum description length (MDL) described as

$$f_{\mathrm{DSM}}(M) = n_M \log(n) + \log(n) * \sum_{i=1}^{n_M} M_i + (1 + 2\log(n))(|S_1| + |S_2|)$$

where n_M is the number of clusters, n is the number of decision variables, M_i is the number of variables in the ith cluster, and S_1 and S_2 present two mismatch sets. Taking Fig. 2 (right) for example, $n_M = 4, n = 9, |S_1| = |S_2| = 0, M_1 = 4, M_2 = 3, M_3 = 2$, and $M_4 = 1$.

By introducing MDL metric, the DSM clustering problem is translated into an optimization problem. The aim of DSM clustering is to search for a clustering arrangement M that minimizes the metric $f_{\mathrm{DSM}}(M)$.

D. Multiobjective Optimization

In this paper, the following continuous MOP [1] is considered:

$$\begin{cases} \min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to} : \mathbf{x} \in \Omega \end{cases} \quad (3)$$

where $\mathbf{F}(\mathbf{x}) : \Omega \to \mathbf{R}^m$ is made up of m real-valued continuous functions which need to be minimized simultaneously. Ω is the feasible decision space. $\mathbf{x}$ is the decision vector within the feasible region Ω.

Decision vector $\mathbf{x}_u$ is said to dominate $\mathbf{x}_v$ (expressed as $\mathbf{x}_u \prec \mathbf{x}_v$)), if $\forall i \in \{1, \dots, m\}, f_i(\mathbf{x}_u) \leq f_i(\mathbf{x}_v)$, and $\mathbf{F}(\mathbf{x_u}) \neq \mathbf{F}(\mathbf{x_v})$. Moreover, $\mathbf{x}_u$ is said to strongly dominate $\mathbf{x}_v$ if $\forall i \in \{1, \dots, m\}, f_i(\mathbf{x}_u) < f_i(\mathbf{x}_v)$. A vector $\mathbf{x}^* \in \Omega$ is known as Pareto optimal if there does not exist a vector $\mathbf{x} \in \Omega$ dominating it. The set of all Pareto-optimal solutions is defined as Pareto-optimal set (PS), i.e., PS $= \{\mathbf{x}^* \mid \neg\exists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}$. The map of the PS is defined as the Pareto-optimal front (PF), PF $= \{\mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in \text{PS}\}$. In engineering, it is usually unnecessary and impractical to find the whole PF of a continuous MOP. Therefore, solving a continuous MOP involves finding a manageable number of Pareto-optimal solutions that are uniformly distributed over the PF.

E. Regularity Property of Continuous MOP

The Karush–Kuhn–Tucker (KKT) condition indicates that under mild conditions, the dimension of the PS for an MOP (3) is $(m-1)$ [32], [33]. This property is called regularity property of continuous MOPs [14]. The following theorem [32], [33] describes this regularity property in detail.

Theorem 1: Suppose the objective functions $f_i(\mathbf{x}), i = 1, \dots, m$ are continuously differentiable at $\mathbf{x}^* \in \Omega$. If $\mathbf{x}^*$ is a (local) Pareto-optimal solution, there exists a vector $\alpha \geq \mathbf{0}$ satisfying

$$\begin{cases} \sum_{i=1}^{m} \alpha_i \nabla f_i(\mathbf{x}^*) = \mathbf{0} \\ \sum_{i=1}^{m} \alpha_i = 1. \end{cases} \quad (4)$$

The points satisfying (4) are known as KKT points. Equation (4) has $n + 1$ equality constraints and $n + m$ variables $x_1, \dots, x_n, \alpha_1, \dots, \alpha_m$. Thus, under mild conditions, the distribution of PS to MOP (3) is a piecewise continuous $(m-1)$-D manifold. Specifically, under mild conditions, the PS is a piecewise continuous curve for a continuous biobjective problem and a piecewise continuous curved surface for a continuous triobjective problem. Most continuous ZDT [34], DTLZ [35], UF and constrained multiobjective objective functions [13], and MOP [36] problems meet this regularity property.

F. Control Property of Decision Variables

In addition to separability, decision variables also have their control property in terms of their relationship with the fitness landscape in MOP. The following types of relationships are interesting because we can use them to separate the spread and convergence parts of the solutions found for an MOP [12]. A way to learn the conflict among objective functions is recognizing the variables which control the diversity of the generated solutions. The first kind of decision variable is called position variable. A decision variable x_i is called position variable [12] if and only if changing x_i in $\mathbf{x} = (x_1, \dots, x_n)$ can only cause a vector that is incomparable or equivalent to $\mathbf{x}$. Changing a position variable on its own never causes a dominated or dominating decision vector.

Instead, if changing x_i in $\mathbf{x} = (x_1, \dots, x_n)$ can only result in a decision vector which equals $\mathbf{x}$, dominates $\mathbf{x}$, or is dominated by $\mathbf{x}$, then x_i is called a distance variable. That is to say, changing a distance variable on its own will never cause incomparable decision vectors.

All decision variables which are neither position nor distance variables are called mixed variables. Furthermore, changing a mixed variable on its own can cause a change in distance or position.

The difference among distance, position and mixed variables have been highlighted in Fig. 3. The used multiobjective optimization problem in Fig. 3 is described as

$$\begin{cases} \min f_1(\mathbf{x}) = x_1 - \frac{1}{2}\cos(3.2\pi x_2) + x_3^2 \\ \min f_2(\mathbf{x}) = 1 - x_1 + \sin(3.2\pi x_2) + x_3^2 \\ \text{subject to} : x_i \in [0, 1], i = 1, 2, 3. \end{cases}$$

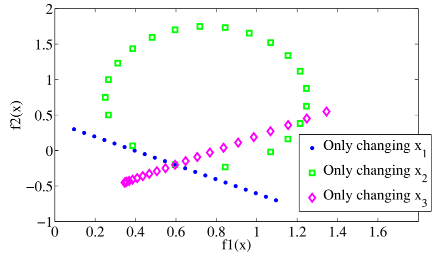


Fig. 3. Plot of the sampling points by changing one variable and the other variables are fixed as 0.5. Where the multiobjective problem is $\min f_1(\mathbf{x}) = x_1 - (1/2)\cos(3.2\pi x_2) + x_3^2$, $\min f_2(\mathbf{x}) = 1 - x_1 + \sin(3.2\pi x_2) + x_3^2$, satisfying $x_i \in [0, 1]$, $i = 1, 2, 3$.

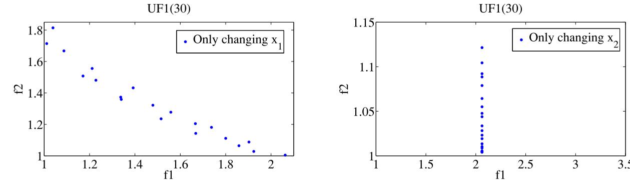


Fig. 4. Plot of the sampling solutions by varying x_1 only (left) and x_2 only (right) for $\mathbf{x} = (0.5, 0.5, \ldots, 0.5)$ on UF1 problem.

For any fixed x_2, x_3, changing the value of x_1 on its own in $\mathbf{x} = (x_1, x_2, x_3)$ results in a set of incomparable or equivalent solutions. Therefore, x_1 is a position variable for the used MOP. For any fixed x_1, x_2, only changing x_3 in $\mathbf{x} = (x_1, x_2, x_3)$ will never result in incomparable solutions. Therefore, x_3 is a distance variable for the used MOP. There exists $x_1 = x_2 = x_3 = 0.5$ such that only changing the value of x_2 in $\mathbf{x} = (x_1, x_2, x_3)$ results in a set of solutions including dominated solutions and nondominated solutions. Therefore, x_2 is a mixed variable for the used MOP.

For continuous ZDT, DTLZ, UF, and MOP [36] problems, the number of position variable(s) and mix variable(s) is $m-1$, while the number of distance variable(s) is $n - m + 1$. Where m is the number of objective functions in MOP (3) and n is the number of decision variables.

III. Proposed Algorithm: MOEA/DVA

First, we analyze the control property of decision variables. Then, Definition 2 is used to learn the variable linkage. Finally, MOEA/DVA is proposed based on the above DVAs.

A. Control Variable Analysis

In MOPs, some decision variables control the convergence aspect of the obtained solutions, while some decision variables determine the spread aspect of the obtained solutions [12]. Inspired by this, it is useful for optimization algorithms to separate decision variables based on their control property (convergence or/and spread). The definition of position variable, distance variable, and mixed variable can be found in Section II-F. The position variables and mixed variables are beneficial to learn the conflict among objective functions, while distance variables have the most important and direct impact on the convergence of evolutionary population. Algorithm 1 gives the detail of control variable analysis.

Taking UF1 for example, x_1 is a mixed variable and $x_2, x_3, \ldots, x_n$ are distance variables. For $\mathbf{x} = (0.5, 0.5, \ldots, 0.5)$, Fig. 4 (left) illustrates the sampling solutions by varying x_1 only, while Fig. 4 (right) plots the sampling solutions by changing x_2 only for UF1 problem with 30 variables. From Fig. 4 (left), we can see that the number of nondominated solutions sampled is greater than 1 and less than NCA (the number of sampling solutions). Therefore, variable x_1 not only controls the spread of generated solutions but also controls their convergence. That is to say x_1 is a mixed variable for UF1. From Fig. 4 (right), we can see that changing x_2 only results in dominating solutions or dominated solutions. There will never exist incomparable solutions in the sampling solutions. Thus, variable x_2 only controls the optimization difficulty of UF1 problem. That is to say, x_2 can be a distance variable for UF1 problem.

When the proposed algorithm performs the control analysis of decision variables, the number of objective function evaluations required is $n \times$ NCA, where NCA is the number of sampling solutions. Based on a large number of samples by Algorithm 1, Table I concludes the control property analysis for the existing benchmark MOPs, such as continuous ZDT, DTLZ, UF1-UF10, and three-objective walking fish group (WFG) problems. In this table, three-objective WFG problems have $n = 24$ variables and the number of position-related variables $k = 4$.

B. Interdependence Analysis Between Two Decision Variables

As discussed in Section II-A, there exist different definitions about interacting variables. In this paper, Definition 2 is used to analyze the interdependence relationship between two decision variables.

1) Two Necessary Conditions for Interdependent Decision Variables: In order to help the reader understand the concept of how two decision variables interact, we provide two necessary conditions for interacting decision variables of continuously differential function. In this paper, the definition of interdependent variables is based on Definition 2 unless stated otherwise.

Theorem 2 (Necessary Condition for Interacting Decision Variables): Suppose $f(\mathbf{x})$ is a continuously differential function. If two decision variables x_i and x_j interact, then $(\partial f(\mathbf{x})/\partial x_i)$ is dependent on x_j.

Proof: Let $(\partial f(\mathbf{x}))/\partial x_i$ be not dependent on $x_j \Rightarrow \forall \mathbf{x} = (x_1, \ldots, x_j, \ldots, x_n), b_1, b_2$, satisfying $(\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_1} = (\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_2} \Rightarrow \forall \mathbf{x} = (x_1, \ldots, x_j, \ldots, x_n), a_1, a_2, b_1, b_2$, satisfying $\int_{a_1}^{a_2} (\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_1} dx_i = \int_{a_1}^{a_2} (\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_2} dx_i \Rightarrow \forall \mathbf{x} = (x_1, \ldots, x_j, \ldots, x_n), a_1, a_2, b_1, b_2$, satisfying $f(\mathbf{x})|_{x_i=a_2, x_j=b_1} - f(\mathbf{x})|_{x_i=a_1, x_j=b_1} = f(\mathbf{x})|_{x_i=a_2, x_j=b_2} - f(\mathbf{x})|_{x_i=a_1, x_j=b_2}$.

According to Definition 2, x_i does not interact with x_j. This contradicts the condition that x_i and x_j are interacting. Therefore, Theorem 2 has been proven. ■

Theorem 2 is beneficial for the reader to judge whether two decision variables are separable. Taking the sphere function

Algorithm 1 Control Variable Analysis

Require: n: number of variables in MOP (3). *FE*: used number of function evaluations.
Ensure: *DiverIndexes*: indexes of diverse variables (position variables and mixed variables).
ConverIndexes: indexes of distance variables.
For $i = 1$ to n
Generate random vector $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_n)$ in its feasible domain. Set $DiverIndexes = ConverIndexes = S = \emptyset$.
For $j = 1$ to *NCA*
$x_i^{'} \longleftarrow x_i^L + \frac{j-1+rand}{NCA} * (x_i^U - x_i^L)$, where *rand* is a random number in [0, 1). x_i^L and x_i^U are respectively the lower bound and upper bound of ith decision variable.
Add $\mathbf{F}(x_1, \ldots, x_{i-1}, x_i^{'}, x_{i+1}, \ldots, x_n)$ into the sample set S and set $FE = FE+1$.
End for
Use nondominated sorting (Deb *et al.*, 2002) for S to obtain different nondominated fronts.
If the size of the first nondominated front is equal to *NCA*, **then** x_i can be a position variable. Add index i to *DiverIndexes*.
Else if the sizes of all the nondominated fronts are equal to 1, **then** x_i can be a distance variable. Add index i to *ConverIndexes*.
Else x_i is a mixed variable. Add index i to *DiverIndexes*.
End for

TABLE I
CONTROL PROPERTY ANALYSIS AND INTERACTION ANALYSIS FOR THE EXISTING BENCHMARK MOPS, WHERE THREE-OBJECTIVE WFG PROBLEMS HAVE $n = 24$ VARIABLES AND THE NUMBER OF POSITION-RELATED VARIABLES $k = 4$. THE OBTAINED RESULTS ON CONTROL PROPERTY OF VARIABLES ARE BASED ON A LARGE NUMBER OF SAMPLING BY ALGORITHM 1. THE OBTAINED RESULTS ON INTERACTION ANALYSIS ARE BASED ON A LARGE NUMBER OF SAMPLING BY ALGORITHM 2

MOP	Control property analysis			Interaction analysis
	Position variable	Mixed variable	Distance variable	
ZDT1,ZDT2,ZDT4,ZDT6	x_1	-	$x_2, \ldots, x_n$	no interaction
ZDT3	-	x_1	$x_2, \ldots, x_n$	no interaction
DTLZ1-DTLZ4	$x_1, \ldots, x_{m-1}$	-	$x_m, \ldots, x_n$	no interaction
DTLZ7	-	$x_1, \ldots, x_{m-1}$	$x_m, \ldots, x_n$	no interaction
UF1-UF10	-	$x_1, \ldots, x_{m-1}$	$x_m, \ldots, x_n$	sparse interactions
WFG1,WFG4,WFG5	$x_1, \ldots, x_4$	-	$x_5, \ldots, x_{24}$	no interaction
WFG2	x_3, x_4	x_1, x_2	$x_5, \ldots, x_{24}$	sparse interactions
WFG3	$x_1, \ldots, x_4$	-	$x_5, \ldots, x_{24}$	sparse interactions
WFG6,WFG9	$x_1, \ldots, x_4$	-	$x_5, \ldots, x_{24}$	Highly dependent interactions
WFG7	$x_1, \ldots, x_4$	$x_5, \ldots, x_{14}$	$x_{15}, \ldots, x_{24}$	Highly dependent interactions
WFG8	x_1, x_2	x_3, x_4	$x_5, \ldots, x_{24}$	Highly dependent interactions

for example, $(\partial f(\mathbf{x})/\partial x_i) = 2x_i, i = 1, \ldots, n$ is not dependent on the other variables. We can solve the sphere function by optimizing decision variables one by one. Therefore, the sphere function is a separable function. For Rosenbrock's function, $(\partial f(\mathbf{x})/\partial x_i) = 400x_i(x_i^2 - x_{i+1}) + 2(1 + x_i) - 200(x_{i-1}^2 - x_i), i = 2, \ldots, n-1$. Therefore, the variable interaction for Rosenbrock's function can only exist between x_i and x_{i+1}, $i = 1, \ldots, n-1$ possibly. The definition of the sphere function and Rosenbrock's function can be found in [15] and [16].

Theorem 3 (Necessary Condition for Interacting Decision Variables): Let $f(\mathbf{x})$ be a continuously differential function. If two decision variables x_i and x_j interact, then $\exists \mathbf{x}, a_1, a_2, b_1$, and b_2 meeting

$$f(\mathbf{x})|_{x_i=a_2,x_j=b_1} - f(\mathbf{x})|_{x_i=a_1,x_j=b_1} \neq f(\mathbf{x})|_{x_i=a_2,x_j=b_2} - f(\mathbf{x})|_{x_i=a_1,x_j=b_2}. \quad (5)$$

Proof: According to Theorem 2, if two decision variables x_i and x_j interact, then $(\partial f(\mathbf{x}))/\partial x_i$ is dependent on x_j. $\Rightarrow$ $\exists \mathbf{x}, a_1, a_2, b_1$, and b_2 satisfying $(\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_1} \neq (\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_2} \Rightarrow \exists \mathbf{x}, a_1, a_2, b_1, b_2$ satisfying $\int_{a_1}^{a_2}(\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_1}dx_i \neq \int_{a_1}^{a_2}(\partial f(\mathbf{x})/\partial x_i)|_{x_j=b_2}dx_i$ $\Rightarrow$ $\exists \mathbf{x} = (x_1, \ldots, x_j, \ldots, x_n), a_1, a_2, b_1, b_2$, meet $f(\mathbf{x})|_{x_i=a_2,x_j=b_1} - f(\mathbf{x})|_{x_i=a_1,x_j=b_1} \neq f(\mathbf{x})|_{x_i=a_2,x_j=b_2} - f(\mathbf{x})|_{x_i=a_1,x_j=b_2}$. ■

It is a pity that the conditions in Theorems 2 and 3 are not sufficient conditions. That is to say, $(\partial f(\mathbf{x})/\partial x_i)$ is dependent on $x_j \not\Rightarrow x_i, x_j$ are interacted. An example is $f(x_1, x_2) = (x_1 + x_2)^3 = x_1^3 + 3x_1^2x_2 + 3x_1x_2^2 + x_2^3$. Due to $(\partial f(\mathbf{x}))/\partial x_1 = (\partial f(\mathbf{x}))/\partial x_2 = 3(x_1 + x_2)^2 \geq 0$, $(\partial f(\mathbf{x}))/\partial x_1$ depends on x_2. However, there does not exist $\mathbf{x}, a_1, a_2, b_1$, and b_2 simultaneously satisfying $f(\mathbf{x})|_{x_1=a_2,x_2=b_1} < f(\mathbf{x})|_{x_1=a_1,x_2=b_1}$ and $f(\mathbf{x})|_{x_1=a_2,x_2=b_2} > f(\mathbf{x})|_{x_1=a_1,x_2=b_2}$ due to $(\partial f(\mathbf{x})/\partial x_1) \geq 0$ for arbitrary x_2. Therefore, the conditions in Theorems 2 and 3 are necessary but not sufficient for the interaction between the two decision variables x_i and x_j.

2) Learning the Interaction Between Two Decision Vectors: Due to the modular nature, real-world optimization problems will most likely consist of a set of independent subcomponents [16], [19]. For such problems, variable linkage detecting method can be effective in decomposing a

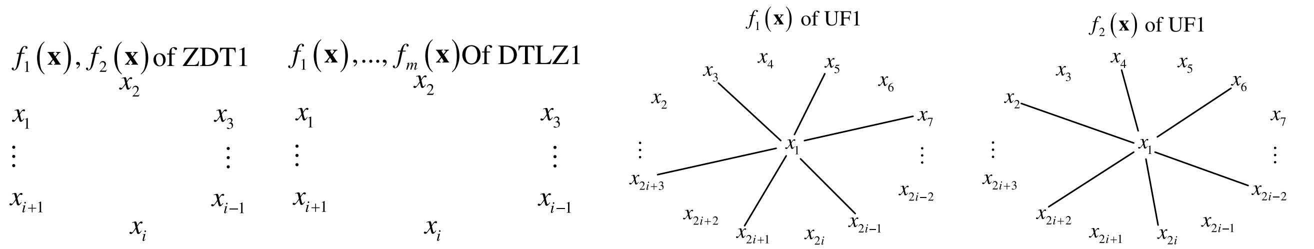


Fig. 5. Interdependence analysis between two decision variables for ZDT1 problem (left), DTLZ1 problem (middle), and biobjective UF1 problem (right). Where the linkage edge indicates that two decision variables interact. The variable interactions are learned based on a large number of samples using Definition 2.

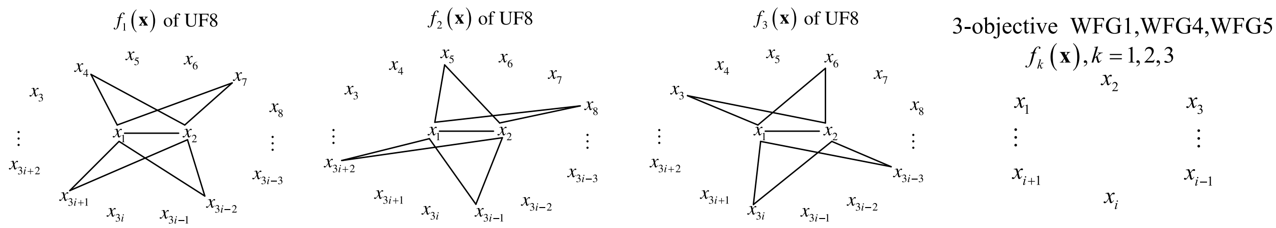


Fig. 6. Interdependence analysis between two decision variables for triobjective UF8 problem (left) and three WFG problems (right). Where the linkage edge indicates that two decision variables interact. The variable interactions are learned based on a large number of samples using Definition 2.

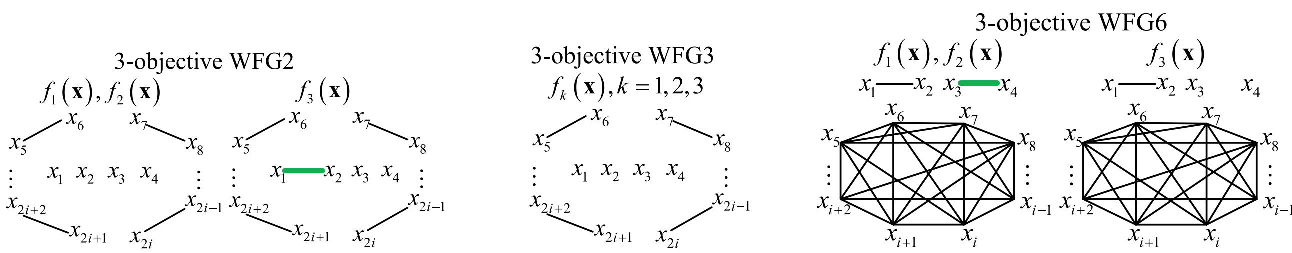


Fig. 7. Interdependence analysis between two decision variables for triobjective WFG problems. Where the linkage edge indicates that two decision variables interact. The variable interactions are learned based on a large number of samples using Definition 2.

high-dimensional vector into a set of low-dimensional subcomponents. The above divide-and-conquer idea can be beneficial for optimizing the problem because it optimizes a group of interdependent variables (a subcomponent) together, rather than all variables. However, the main difficulty lies in how to decompose the decision vector into a set of subcomponents. Without any *a priori* knowledge on the structure of the given problem, the problem at hand can be decomposed by many various ways.

In fact, the interactions among decision variables can be utilized to decompose a difficult function into a set of subfunctions with low-dimensional subcomponents [21]. Tezuka *et al.* [24] used formula (5) without derivation to learn the interaction between two decision variables in the process of evolution. For additively separable functions, Omidvar *et al.* [21] gave a theoretical derivation for using formula (5) to recognize the interacting decision variable. However, formula (5) is a necessary but not sufficient condition to recognize that two decision variables interact for continuously differential function as described in Section III-B1. In this paper, we suggest to use Definition 2 to learn the interaction relationship between two decision variables. Algorithm 2 gives the detail of learning the interaction between two decision variables. Figs. 5–7, respectively, plot the interacting relationship between two decision variables for ZDT1 and DTLZ1, UF1 and UF8, and five WFG problems. The variable interactions are learned based on a large number of samples using Definition 2. There are two outstanding features for these problems. One is no variable interaction existing in individual functions of ZDT1, DTLZ1, WFG1, and WFG4–WFG5 problems. The other is that sparse variable interactions focus on $m-1$ decision variables for UF1 and UF8 problems, where m is the number of objective functions defined in (3). Generally, according to Definition 2, there are sparse variable interactions for most benchmark test problems including continuous ZDT, DTLZ, UF, and WFG problems. Furthermore, according to Definition 1, Appendix A of supplementary material A provides the proof that objective functions $f_k(\mathbf{x})$, $k=1,\ldots,m$ in most continuous ZDT and DTLZ problems are separable functions.

Taking UF1 for example, x_1 has interaction with $x_3, x_5, \ldots$ for $f_1(\mathbf{x})$ while x_1 has interaction with $x_2, x_4, \ldots$ for $f_2(\mathbf{x})$ as shown in Fig. 5.

When performing one judgment of interaction between two variables, the proposed algorithm needs to evaluate the values of objective functions at three points. Therefore, the number

Algorithm 2 Interdependence Analysis Between Two Decision Variables

Require: m: number of objective functions defined in MOP (3). n: number of variables in MOP (3).
$\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ and $\{\mathbf{F}^1, \ldots, \mathbf{F}^N\}$: current evolutionary population and their objective values.
ConverIndexes: index set of distance variables. *FE*: used number of function evaluations.
Ensure: the learned variable linkages.
For $i = 1$ to $n-1$
For $j = i+1$ to n
While time to try $< NIA$ **do**
Randomly select an individual $\mathbf{x}^l$ from population $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ and uniformly random sampling a_2, b_2 in their feasible domain, where $a_1 = x_i^l, b_1 = x_j^l$. Evaluate $\mathbf{F}(\mathbf{x}^l)|_{x_i=a_2}$, $\mathbf{F}(\mathbf{x}^l)|_{x_j=b_2}$, $\mathbf{F}(\mathbf{x}^l)|_{x_i=a_2,x_j=b_2}$, where $\mathbf{F}(\mathbf{x}^l)|_{x_i=a_2,x_j=b_2} = \mathbf{F}(x_1^l, \ldots, x_{i-1}^l, a_2, x_{i+1}^l, \ldots, x_{j-1}^l, b_2, x_{j+1}^l, \ldots, x_n^l)$. Set *FE*=*FE*+3.
For $k = 1$ to m //learn the variable linkage based on Definition 2
$\Delta_i f_k|_{x_j=b_1} \longleftarrow f_k(\mathbf{x}^l)|_{x_i=a_2,x_j=b_1} - f_k(\mathbf{x}^l)|_{x_i=a_1,x_j=b_1}$
$\Delta_i f_k|_{x_j=b_2} \longleftarrow f_k(\mathbf{x}^l)|_{x_i=a_2,x_j=b_2} - f_k(\mathbf{x}^l)|_{x_i=a_1,x_j=b_2}$,
where $f_k(\mathbf{x}^l)|_{x_i=a_2,x_j=b_2} = f_k(x_1^l, \ldots, x_{i-1}^l, a_2, x_{i+1}^l, \ldots, x_{j-1}^l, b_2, x_{j+1}^l, \ldots, x_n^l)$
If $\Delta_i f_k|_{x_j=b_1} * \Delta_i f_k|_{x_j=b_2} < 0$, **then** detect interaction between x_i and x_j for $f_k(\mathbf{x})$.
End for
//Use the offspring $\mathbf{x}^l|_{x_i=a_2}$, $\mathbf{x}^l|_{x_j=b_2}$, $\mathbf{x}^l|_{x_i=a_2,x_j=b_2}$ to update the parent $\mathbf{x}^l$
If $j \in$ *ConverIndexes* & $\mathbf{F}(\mathbf{x}^l)|_{x_j=b_2} \prec \mathbf{F}^l$, **then** $x_j^l = b_2, \mathbf{F}^l = \mathbf{F}(\mathbf{x}^l)|_{x_j=b_2}$
If $i \in$ *ConverIndexes* & $\mathbf{F}(\mathbf{x}^l)|_{x_i=a_2} \prec \mathbf{F}^l$, **then** $x_i^l = a_2, x_j^l = b_1, \mathbf{F}^l = \mathbf{F}(\mathbf{x}^l)|_{x_i=a_2}$
If $i, j \in$ *ConverIndexes* & $\mathbf{F}(\mathbf{x}^l)|_{x_i=a_2,x_j=b_2} \prec \mathbf{F}^l$, **then** $x_i^l = a_2, x_j^l = b_2, \mathbf{F}^l = \mathbf{F}(\mathbf{x}^l)|_{x_i=a_2,x_j=b_2}$
End while
End for
End for

Algorithm 3 Dividing Distance Variables Based on Two Variable Analyses

Require: m: the number of objective functions defined in (3). n: the number of variables in MOP (3).
Ensure: A set of subcomponents, $pop = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ and $Obj = \{\mathbf{F}^1, \ldots, \mathbf{F}^N\}$.
Step 1: Learning the interaction: Use Algorithm 2 to obtain variable linkages for each objective function.
Step 2: Divide distance variables based on maximal connected subgraphs of the variable linkage graph.

of objective function evaluations required for interdependence analysis is $(3/2)n(n-1)*$NIA, where m is the number of objective functions, n is the number of decision variables defined in MOP (3), and NIA is the maximum number of tries required to judge the interaction between two variables. The larger NIA is, the more precise the judgment of interaction between two variables will be.

C. Dividing the Distance Variables

As mentioned above, it is interesting to decompose the objective function $f_k(\mathbf{x}), k = 1, \ldots, m$ with high-dimension variables into a set of subfunctions with low-dimensional subcomponents. Ideally, the subcomponents should be formed according to the interaction of the decision variables so that the interactions between the subcomponents are kept to a minimum [21]. Algorithm 3 gives the detail of the dividing method for distance variables based on variable linkages in all objective functions.

In step 2 of Algorithm 3, MOEA/DVA just decomposes distance variables. The idea behind it is that we want to separate the optimization difficulty of MOP and the conflict among objective functions. The distance variable plays an important role in the convergence of the population, while position variable plays an important role in the diversity of the population. Therefore, the distance variables are the main optimization difficulty of MOP, while the position variables are the main conflict among objective functions. We fix the values of diverse variables (position variables and mixed variables) and evolve distance variables only in the early stages of evolution.

Different from single objective optimization problem, MOP needs to optimize all objective functions together. Therefore, it needs to divide decision variables considering the variable linkages in all objective functions. We synthesize the variable linkages in each objective function into one graph of variable linkage. Fig. 8 takes a three-objective problem to illustrate the process of dividing variables based on maximal connected subgraph. The final subcomponents are $\{x_3, x_4, x_5, x_6\}$, $\{x_7\}$, and $\{x_8, x_9, x_{10}\}$. x_1 and x_2 are not present in the subcomponents because they are diverse variables fixed during the early stages of evolution. The reason why x_3 and x_6 belong to the same subcomponent is that x_3 and x_6 belong to the same maximal connected subgraph ($x_3 \overset{f_1}{\text{—}} x_4 \overset{f_2}{\text{—}} x_5 \overset{f_3}{\text{—}} x_6$). But x_6 and x_8 belong to different subcomponents because they belong to different maximal connected subgraphs.

Taking continuous ZDT, DTLZ, and UF1–UF10 problems, for example, the final subcomponents are $\{x_m\}, \ldots, \{x_n\}$ by using Algorithm 3. Each subcomponent has only one variable $x_i, i = m, \ldots, n$.

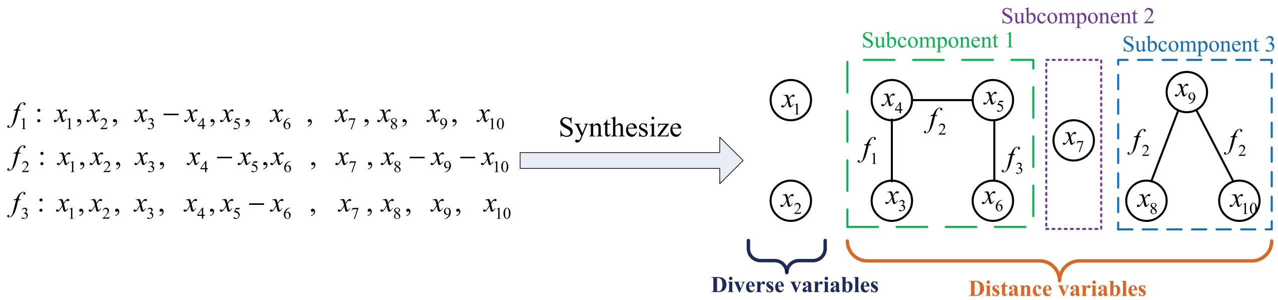


Fig. 8. Taking three-objective problem to show how to divide distance variables for MOP based on the variable linkages in all of objective functions. Left: linkage edge indicates that two decision variables interact. The variable linkages are learned based on sampling by using Definition 2.

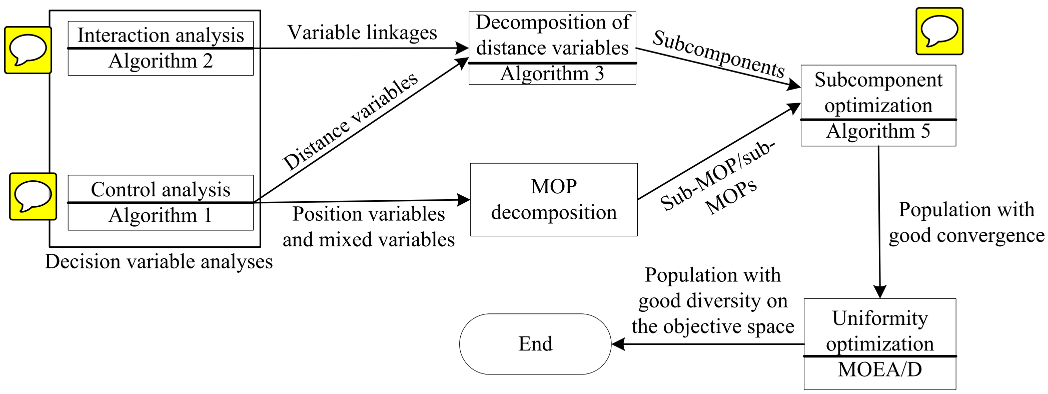


Fig. 9. Algorithmic idea and flow chart of MOEA/DVA.

The defect of Algorithm 3 is that decision variables having direct or indirect linkage will be grouped into the same subcomponent. The size of a subcomponent may be large if there are many linkages between decision variables. There are other algorithms that try to break the linkages and make the subcomponents smaller. Why does our proposed algorithm not ignore some variable linkages and make the subcomponents smaller? Based on Definition 2, we may judge whether the two variables have linkage or not. However, the proposed algorithm does not know the correlation between two variables. Therefore, in the current version of MOEA/DVA, we cannot estimate the influence by ignoring some linkages. EDA may be the promising technology to detect the linkage degree among decision variables.

D. Framework of the Proposed MOEA/DVA

The algorithmic idea and flow chart of MOEA/DVA are shown in Fig. 9. The process of MOEA/DVA for optimization can be concluded as follows.

1) *Decision Variable Analyses:* There are two variable analyses: a) control property analysis and b) interaction analysis. Interaction analysis provides the variable linkages for decomposition of distance variables, while control property analysis provides diverse variables (position variables and mixed variables) for MOP decomposition and offers distance variables for decomposition of distance variables.
2) *Decomposition of Distance Variables:* Decompose high-dimensional distance variables into several low-dimensional subcomponents which can be optimized more easily.
3) *MOP Decomposition Based on Diverse Variables:* A MOP is decomposed into a set of sub-MOPs with uniformly distributed values of diverse variables (position variables and mixed variables).
4) *Subcomponent Optimization:* Optimize each subcomponent independently to improve the convergence speed of population.
5) *Uniformity Optimization:* Optimize all the decision variables including position variables and mixed variables. Its aim is to improve the uniformity of population in the objective space.

The above five techniques are introduced one by one below. As shown in Fig. 9, it is clear that the basis of MOEA/DVA is the decision variable analyses. Two DVAs are respectively described in Sections III-A and III-B. The motivation of DVAs is to discover the potential/hidden features of decision variables. These potential/hidden features (variable dependencies, control property of decision variables, and so on) will be beneficial to solving the MOP.

The core of MOEA/DVA lies in two kinds of decompositions including decomposition of distance variables and MOP decomposition based on diverse variables. Decomposition of distance variables is based on variable linkages learned by the interdependence analysis introduced in Section III-B. The detail of decomposing distance variables is described in Algorithm 3 of Section III-C. MOEA/DVA divides distance variables based on maximal connected subgraph as shown in Fig. 8 (right).

TABLE II
DIFFERENCES BETWEEN SUB-MOP, SUBPROBLEM, AND MULTIOBJECTIVE SUBPROBLEM

	Sub-MOP in MOEA/DVA	Subproblem in MOEA/D	Multiobjective subproblem in MOEA/D-M2M
Scope	a MOP with a fixed value of diverse variables	single objective optimization problem	a MOP with constrained objective space
Feature	diverse variable(s)	weight vector	preference direction
Optimization	Based on variables decomposition	Treat all the decision variables as a whole to optimize	Treat all the decision variables as a whole to optimize

$$\begin{cases} f_1(\mathbf{x}) = x_1 + 2\left[x_3 - \sin\left(6\pi x_1 + \frac{1}{3}\pi\right)\right]^2 \\ f_2(\mathbf{x}) = 1 - (x_1)^{0.5} + 2\left[x_2 - \sin\left(6\pi x_1 + \frac{2}{3}\pi\right)\right]^2 \\ s.t. \quad x_1 \in [0,1], x_2, x_3 \in [-1,1] \end{cases} \qquad \begin{cases} f_1(\mathbf{x}) = 0.25 + 2\left[x_3 - \sin\left(1.5\pi + \frac{1}{3}\pi\right)\right]^2 \\ f_2(\mathbf{x}) = 0.5 + 2\left[x_2 - \sin\left(1.5\pi + \frac{2}{3}\pi\right)\right]^2 \\ s.t. \quad x_2, x_3 \in [-1,1] \end{cases}$$

$$\text{Original MOP} \xrightarrow[\text{Value of diverse variables is fixed}]{x_1 = 0.25} \text{Derived sub-MOP}$$

Fig. 10. Illustrate the relationship between original MOP and its sub-MOP.

Decomposing an MOP is more difficult than decomposing a single objective problem. In MOPs, there is no unique solution in general. Changing the decision vector will result in incomparable solutions. Therefore, it is necessary to analyze the conflicts among objective functions. A way to learn the conflicts among objective functions is to recognize the variables which control the diversity of the generated solutions. Therefore, position variables and mixed variables are used to decompose a complex MOP.

Different from the decomposition in MOEA/D [2] based on weight vector and the decomposition in MOEA/D-M2M [36] based on preference direction, this paper uses diverse variables (position variables and mixed variables) to decompose a difficult MOP (3) into a set of simpler sub-MOPs whose diverse variables are uniformly distributed. The differences among sub-MOP, subproblem, and multiobjective subproblem are listed in Table II. Each sub-MOP is a multiobjective optimization problem which is defined by the original MOP (3) with a fixed value of diverse variables. UF1 problem with three decision variables is taken as an example to explain the concept of sub-MOP. According to the control analysis in Section III-A, x_1 of UF1 problem is a mixed variable and x_2 and x_3 are distance variables. The original MOP is plotted on Fig. 10 (left) while a sub-MOP with constant value of diverse variable $x_1 = 0.25$ is illustrated in Fig. 10 (right). The main feature of sub-MOP is that it only has distance variables without diverse variables (position variables and mixed variables).

The structure of evolutionary population is illustrated in Fig. 11, where N is the population size. In this paper, MOEA/DVA optimizes single evolutionary population and all subcomponents share the same population. Each individual in the population represents a sub-MOP. In this figure, we suppose that x_1 and x_2 are diverse variables (position variables or mixed variables), while $x_3, x_4, \ldots, x_8$ are distance variables. The distance variables are divided into three independent subcomponents $\{x_3\}$, $\{x_4, x_5, x_6\}$, and $\{x_7, x_8\}$. The proposed algorithm fixes the values of diverse variables of

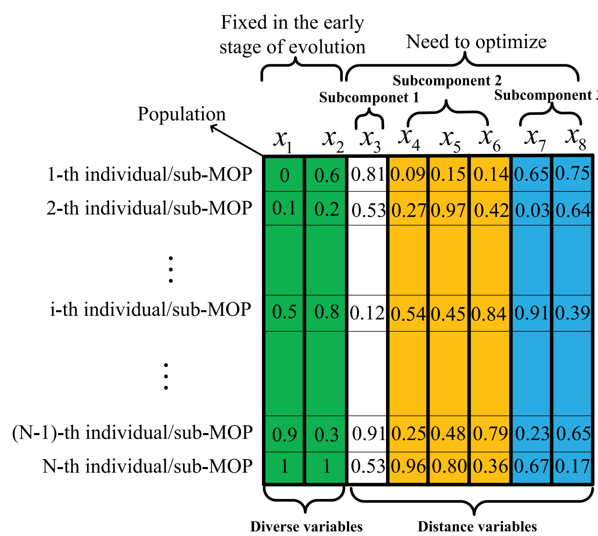


Fig. 11. Structure of population used by MOEA/DVA.

the population and evolves distance variables only in the early stages of evolution. One of the features of sub-MOP is that its value of diverse variables is fixed in the early stages of evolution. Hence, the distribution of values of diverse variables of population has an important effect on the distribution of the obtained solutions. To keep the diversity of evolutionary population, uniformly sampling method [37] is used to initialize the values of diverse variables of the population.

Different from single objective problems, multiobjective problems generally have conflicting objective functions. The conflicts among objective functions refer to incomparable solutions to be generated by changing decision variables. First, position variables and mixed variables have an important effect on the spread of generated solutions. In order to handle the conflicts among objective functions, an MOP is decomposed into a set of sub-MOPs based on diverse variables. In this paper, this decomposition is known as MOP decomposition based on diverse variables. Second, distance variables play an important role in the convergence of

TABLE III
DIFFERENCES BETWEEN MOP DECOMPOSITION BASED ON DIVERSE VARIABLES AND DECOMPOSITION OF DISTANCE VARIABLES

	MOP decomposition based on diverse variables	Decomposition of distance variables
Aim	Handle the conflicting among objective functions	Reduce the optimization difficulty of MOP
Used technique	Control property analysis of variables	Interaction analysis between two variables
Advantage	A complex MOP $\rightarrow$ a set of simpler sub-MOPs	A sub-MOP with high-dimensional distance variables $\rightarrow$ a set of problem with low-dimensional component
Feature of variables	The value of diverse variables of sub-MOP is fixed in the early stage of evolution	Each distance variable of sub-MOP needs to optimize

Algorithm 4 Proposed MOEA/DVA

Require: m: number of objective functions defined in MOP (3). n: number of variables in MOP (3).
N: size of evolutionary population. FE: the used number of function evaluations.
FE_{max}: the maximal number of function evaluations.

Ensure: the optimized population *pop* and their objective values *Obj*.

1. $FE = 0$. Use Algorithm 1 to analyze the control property of variables. Let *DiverIndexes* be the index set of position variables and mixed variables and *ConverIndexes* be the index set of distance variables.
2. $pop(:, DiverIndexes) \longleftarrow$ use uniformly sampling method [37] to initialize the diverse variables of population.
3. $pop(:, ConverIndexes) \longleftarrow$ randomly initialize the distance variables of population.
4. Evaluate these solutions $Obj \longleftarrow \mathbf{F}(pop)$, $OldObj \longleftarrow Obj$ and set $FE = FE + N$.
5. $[Subcomponents, pop, Obj] \longleftarrow$ use Algorithm 3 to divide distance variables and evolve population $[pop, Obj]$, where $\{\mathbf{x}^1, \ldots, \mathbf{x}^N\} = pop$ and $\{\mathbf{F}^1, \ldots, \mathbf{F}^N\} = Obj$.
6. **If** $(m{=}{=}2)$ $threshold \longleftarrow 0.01$.
   Else $threshold \longleftarrow 0.04$.
7. $gen \longleftarrow 0$. $utility \longleftarrow 1$.
8. **While** $utility \geq threshold \bigwedge FE < FE_{max}$ **do**
9.   **For** $j = 1$ to size(*Subcomponents*)
10. $indexes \longleftarrow Subcomponents[j]$
11. $[pop, Obj] \longleftarrow SubcomponentOptimizer(pop, Obj, indexes)$
12.   **End for**
13. $gen = gen + 1$.
14.   If $(gen\%2 == 0)$ $utility \longleftarrow$ CalculateUtilityofSubcomponentOptimization $(Obj, OldObj)$.
15. **End While**
16. **While** $FE < FE_{max}$ **do**
17.   Use one of existing MOEAs [2]–[4] to evolve $[pop, Obj]$. Here MOEA/D [2] is suggested to evolve $[pop, Obj]$.
18. **End While**

generated solutions. In order to reduce the difficulty of optimization, high-dimensional distance variables are decomposed into a set of low-dimensional subcomponents. Table III lists the differences between MOP decomposition based on diverse variables and decomposition of distance variables. Based on learned variable linkages, MOEA/DVA decomposes distance variables into a set of low-dimensional subcomponents by Algorithm 3. Algorithm 4 provides the detail of MOEA/DVA. In MOEA/DVA, these two decompositions are cooperative to solve MOP. MOEA/DVA first decomposes the difficult MOP into a number of simpler sub-MOPs based on diverse variables with uniformly distributed values. Then each sub-MOP optimizes subcomponents one by one in the early stages of evolution.

Line 5 of Algorithm 4 divides the distance variables. Lines 9–12 of Algorithm 4 is similar to cooperative coevolution framework [7], [21]. Line 11 of Algorithm 4 performs subcomponent optimization introduced in the next paragraph. For simplicity, we assign the same computing resources for each subcomponent in MOEA/DVA. Different computing resource can also be assigned for different subcomponents based on their recent performances [38].

For the subcomponent optimization, we use the evolutionary operator in MOEA/D [2]. Due to each objective function $f_i(\mathbf{x}, i = 1, \ldots, m)$ being continuous, the optimal solutions of neighboring sub-MOPs should be close to one another. Therefore, any information about its neighboring sub-MOPs will be helpful for optimizing the current sub-MOP [2]. Neighborhood relations among these sub-MOPs are defined based on the Euclidean distances between their diverse variables. The ith sub-MOP is a neighbor of the jth sub-MOP if the diverse variables of the ith sub-MOP is close to that of the jth sub-MOP. Algorithm 5 provides the detail of subcomponent optimization. In step 3, due to value of diverse variables to be fixed for each sub-MOP in the early stages of evolution, MOEA/DVA just uses the offspring of ith sub-MOP to update the current solution of ith sub-MOP. For simplicity, Algorithm 5 assigns same computing resources for each

Algorithm 5 [pop, Obj] = SubcomponentOptimizer(*pop, Obj, indexes*)

Require: *pop* and *Obj*: current evolutionary population and their objective values. N: size of population.
indexes: indexes of distance variables in the same subcomponent. *FE*: used number of function evaluations.
Ensure: optimized population *pop* and their objective values *Obj*.
For $i = 1$ to N
1. **Reproduction:** Randomly select two individuals/sub-MOPs, k-th individual/sub-MOP and l-th individual/sub-MOP, from its neighbor individuals/sub-MOPs. Use differential evolution (DE) [39] to generate new variables $\mathbf{y}' \longleftarrow pop(i, indexes) + F * (pop(k, indexes) - pop(l, indexes))$ and then perform mutation operator on $\mathbf{y}'$ with probability p_m.
2. **Repair and evaluate:** If a component of $\mathbf{y}'$ is out of the boundary, its value will be reset by a random value inside the boundary. Let $\bar{\mathbf{y}}$ be the repaired solution and set $\mathbf{y}(i, :) \longleftarrow pop(i, :)$ and $\mathbf{y}(i, indexes) \longleftarrow \bar{\mathbf{y}}$. Then evaluate the new solution $\mathbf{y}$ and set $FE = FE + 1$.
3. **Update of the solution:** If $\sum_{j=1}^{m} f_j(\mathbf{y}) < \sum_{j=1}^{m} Obj(i, j)$, then set $pop(i, :) = \mathbf{y}$ and $Obj(i, :) = \mathbf{F}(\mathbf{y})$.

End for

Algorithm 6 *Utility* $\longleftarrow$ CalculateUtilityofSubcomponentOptimization (*Obj,OldObj*)

Require: *Obj*: the objective values of current evolutionary population.
OldObj: objective values of recent evolutionary population. N: size of evolutionary population.
Ensure: Recent utility of subcomponent optimization.
1. $utility \longleftarrow 0$
2. **For** $i \longleftarrow 1$ to N **do** $utility \longleftarrow utility + \frac{\sum_{j=1}^{m} [OldObj(i,j) - Obj(i,j)]}{N}$
3. $OldObj \longleftarrow Obj$

individual/sub-MOP in MOEA/DVA in the early stage of evolution. More intelligent version of MOEA/DVA will be future work discussed in the conclusion.

Finally, we introduce the necessity of uniformity optimization in MOEA/DVA in line 17 of Algorithm 4. As mentioned above, MOEA/DVA first decomposes an MOP into a set of sub-MOPs with uniformly distributed diverse variables and each sub-MOP optimizes subcomponents one by one. By using uniformly fixed values of diverse variables during the early stages of evolution, MOEA/DVA keeps the diversity of population on diverse variables (in decision space). Therefore, the distribution of found solutions by MOEA/DVA is highly dependent on the mapping of the problem from PS to PF.

In order to deal with this issue, MOEA/D is used to evolve all decision variables including diverse variables in the late stages of evolution (utility $<$ threshold). Its aim is to improve the uniformity of population in objective space. Therefore, the idea of MOEA/DVA is to optimize subcomponents one by one to allow the evolutionary population to have good convergence in the early stages of evolution (utility $\geq$ threshold). In the late stages of evolution, MOEA/DVA optimizes all the decision variables including diverse variables to make the evolutionary population have good uniformity in the objective space. The utility of subcomponent optimization is calculated in Algorithm 6.

In Section IV-C, we take nondominated sorting genetic algorithm (NSGA)-II [40] for example to show how to integrate the proposed mechanism into existing MOEAs.

E. Discussions

In MOEA/DVA, a complicated MOP is first decomposed into a set of simpler sub-MOPs with uniformly distributed values of diverse variables. The distance variables are divided into several low-dimensional subcomponents based on learning variable linkages as shown in Fig. 8. Then each sub-MOP independently optimizes subcomponents one by one. The core of MOEA/DVA are two kinds of decompositions: 1) decomposition of distance variables into a set of low-dimensional subcomponents and 2) MOP decomposition based on diverse variables with uniformly distributed values. One may wonder why do we treat diverse variables and distance variables differently? The reason is that we want to separate the conflict among objective functions and the optimization difficulty of MOP.

To gain a better understanding of why and how well the two decompositions work, the following discussion can give some answers.

1) MOP Decomposition Based On Diverse Variables: The reader may be interested in the reason why MOEA/DVA can just evolve distance variables and assign diverse variables with uniformly distributed values in the early stages of evolution? We will discuss its advantages and disadvantages from different views.

First, under some conditions, diverse variables including mixed variables play an important role in PF/PS. Reader interested in its detail can see Theorem 6 and Corollary 1.

Second, the effectiveness of two kinds of decompositions on the benchmark test MOPs is discussed. We compare PS of the original MOP with the solutions found by MOEA/DVA by just evolving the distance variables as shown in Table IV. In MOEA/DVA, we divide the decision variables into diverse variables and distance variables. So we also project the PS onto the diverse variables and distance variables as shown in the second and third columns of Table IV. Taking ZDT1 problem for example, the projection of its PS on diverse variable x_1 is $[0, 1]$ and the projection of its PS on distance variables $(x_2, \ldots, x_n)$ are $\{(x_2, \ldots, x_n) | x_i = 0, i = 2, \ldots, n\}$.

TABLE IV
RELATIONSHIP BETWEEN PS OF ORIGINAL MOP AND THE INDIVIDUALS/SOLUTIONS FOUND BY MOEA/DVA IN THE STAGE OF SUBCOMPONENT OPTIMIZATION

MOP	PS of original MOP		Solutions found by MOEA/DVA in the early stages of evolution		The relation between PS and the solutions found by MOEA/DVA
	Projection on Diverse variables	Projection on Distance variables	Diverse variables	Pareto optimal solution(s) of sub-MOP	
ZDT1,ZDT2,ZDT6	$x_1 \in [0,1]$	$x_i = 0, i = 2, ..., n$	$x_1 \in [0,1]$	$x_i = 0, i = 2, ..., n$	=
ZDT3	$x_1 \in$ subset of $[0,1]$	$x_i = 0, i = 2, ..., n$	$x_1 \in [0,1]$	$x_i = 0, i = 2, ..., n$	subset
ZDT4	$x_1 \in [0,1]$	$x_i = 0.5, i = 2, ..., n$	$x_1 \in [0,1]$	$x_i = 0.5, i = 2, ..., n$	=
DTLZ1-DTLZ4	$x_i \in [0,1]$, $i = 1, ..., m-1$	$x_i = 0.5, i = m, ..., n$	$x_i \in [0,1]$, $i = 1, ..., m-1$	$x_i = 0.5, i = m, ..., n$	=
DTLZ7	$x_i \in$ subset of [0,1], $i = 1, ..., m-1$	$x_i = 0, i = m, ..., n$	$x_i \in [0,1]$, $i = 1, ..., m-1$	$x_i = 0, i = m, ..., n$	subset
UF1-UF4, UF7,UF8,UF10	$x_i \in [0,1]$, $i = 1, ..., m-1$	$x_i = \varphi_i(x_1, ..., x_{m-1})$, $i = m, ..., n$	$x_i \in [0,1]$, $i = 1, ..., m-1$	$x_i = \varphi_i(x_1, ..., x_{m-1})$, $i = m, ..., n$	=
UF5,UF6,UF9	$x_i, \in$ subset of $[0,1]$, $i = 1, ..., m-1$	$x_i = \varphi_i(x_1, ..., x_{m-1})$, $i = m, ..., n$	$x_i \in [0,1]$, $i = 1, ..., m-1$	$x_i = \varphi_i(x_1, ..., x_{m-1})$, $i = m, ..., n$	subset
3-objective WFG1, WFG4-WFG6	$x_i \in [0, 2i]$, $i = 1, ..., 4$	$x_i = 0.7 \times i$, $i = 5, ..., n$	$x_i \in [0, 2i]$, $i = 1, ..., 4$	$x_i = 0.7 \times i$, $i = 5, ..., n$	=
3-objective WFG2,WFG3	$x_i \in$ subset of $[0, 2i]$, $i = 1, ..., 4$	$x_i = 0.7 \times i$, $i = 5, ..., n$	$x_i \in [0, 2i]$, $i = 1, ..., 4$	$x_i = 0.7 \times i$, $i = 5, ..., n$	subset
3-objective WFG8,WFG9	Unknown	Unknown	$x_i \in [0, 2i]$, $i = 1, ..., 4$	Unknown	Unknown

For sub-MOP of ZDT1 problem with diverse variable $x_1 \in [0, 1]$, the individual/solution found by MOEA/DVA on distance variable is $\{(x_2, \ldots, x_n) | x_i = 0, i = 2, \ldots, n\}$. Therefore, PS of ZDT1 problem is equal to the possible solution set found by MOEA/DVA just evolving distance variables. The same analysis can be suited to ZDT2, ZDT4, ZDT6, DTLZ1–DTLZ4, UF1–UF4, UF7-UF8, UF10, WFG1, and WFG4–WFG6 problems as shown in Table IV. For ZDT3 problem, the projection of PS on diverse variable x_1 is a subset, not all, of its feasible region $[0, 1]$. Therefore, some individuals/solutions found by MOEA/DVA by just evolving distance variables are not Pareto-optimal solutions. PS of ZDT3 problem is a subset of the possible solution set found by MOEA/DVA just evolving distance variables. The same analysis can be used for DTLZ7, UF5–UF6, UF9, and WFG2–WFG3 problems as shown in Table IV. Someone may ask why do MOEA/DVA just evolve distance variables and why is its individual/solution not a Pareto-optimal solution? On the selected test MOPs, the reason comes from two aspects: a) these MOPs have mixed variable(s) and b) PFs of these MOPs are discontinuous or degenerate fronts [12].

There is an obvious feature that for most benchmark MOPs, the projection of PS on distance variables is equal to the Pareto-optimal solutions of all sub-MOPs as shown in the third and fifth columns of Table IV. This means that by evolving distance variables only, MOEA/DVA also has the ability of convergence on most benchmark test problems. MOEA/DVA independently optimizes subcomponents one by one. Thus, its convergence can be improved.

Third, according to the definition of mixed variable introduced in Section II-F, a mixed variable not only has an effect on the diversity of generated solutions but also has an effect on the convergence of generated solutions. In the early stages of evolution, MOEA/DVA fixes the values of position variables and mixed variables of population. This treatment has two defects. One is that the found solution of individual/sub-MOP may not be a Pareto-optimal solution. The reason is that MOEA/DVA treats the mixed variables as position variables. The other is the uniformity of obtained solutions may not be good. In order to alleviate the above two defects, MOEA/DVA optimizes all decision variables including mixed variables on the late stages of evolution (utility $\leq$ threshold). Therefore, the idea behind MOEA/DVA is to optimize distance variables only to make the algorithm converge as fast as possible. The mixed variables are related to both diversity and convergence of the algorithm. We sample mixed variables and position variables uniformly at first for sake of simplicity, and then optimize mixed variables and position variables by using MOEA/D after the algorithm converges to a certain degree.

Fourth, as shown in Table I, the existing benchmark MOPs, such as continuous ZDT, DTLZ, UF1–UF10, and most WFG problems, have several features: a) having no or sparse linkages and b) many distance variables and few diverse variables. Why do the designers of the existing benchmark MOPs tend to design MOPs with many distance variables and few diverse variables? The reason is that the PSs and PFs of these MOPs are easier to describe. If an MOP has many mixed variables, its PS and PF may be more difficult to present. In real-world problems, MOPs may contain many mixed variables and their PSs and PFs tend to be unknown in advance or hard to obtain. For the MOPs with many mixed variables, the current version of MOEA/DVA may not work.

Fifth, how to deal with mixed variables better is an open problem. It will be our future work.

2) Decomposition of Distance Variables into Several Low-Dimensional Subcomponents:

Lemma 1: If an MOP $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, $\mathbf{x} \in \Omega$ meets the following two conditions.

a) All decision variables $x_1, \ldots, x_n$ are distance variables.
b) All distance variables are independent of one another for each objective function $f_i(\mathbf{x})$, $i = 1, \ldots, m$.

Then:

a) the decision variables of this MOP can be independently optimized one by one;
b) this MOP only has one Pareto-optimal solution in the objective space.

TABLE V
DIFFERENCES BETWEEN THEOREMS AND COROLLARY IN SECTION III-E2

	MOP/Sub-MOP	Independence requirement
Theorem 3.4	Sub-MOP	Variable is independent
Theorem 3.5	Sub-MOP	Subcomponent is independent
Theorem 3.6	MOP	Subcomponent is independent
Corollary 3.7	MOP	Variable is independent

The proof of Lemma 1 can be found in Appendix B of supplementary material A. In general, an MOP has a set of Pareto-optimal solutions. Therefore, someone may wonder what the role of this lemma is. To some extent, this lemma suggests two points.

a) MOP usually has position variable(s) or mixed variable(s).
b) Some variables interact with one another.

Moreover, this lemma is served for the optimization of sub-MOP. By fixing the value of diverse variables, the original MOP becomes a sub-MOP with distance variables only as shown in Fig. 10. Lemma 1 suggests that if all distance variables are independent of one another for each objective function, each sub-MOP can optimize variables one by one as described in the following Theorem 4.

Theorem 4: For an MOP $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, $\mathbf{x} \in \Omega$, if the following condition is met.

All distance variables are independent of one another for each objective function $f_i(\mathbf{x})$, $i = 1, \ldots, m$.

Then, we have:

a) each sub-MOP of this MOP only has one Pareto-optimal solution in the objective space;
b) the distance variables of sub-MOP can be independently optimized one by one.

The proof of Theorem 4 can be found in Appendix B of supplementary material A. Taking Fig. 10 for example, the unique Pareto-optimal solution of sub-MOP with $x_1 = 0.25$ is $[0.25, 0.5]$ in the objective space and decision variables x_2, x_3 can be independently optimized one by one for UF1 problem. One thing to be noted is that the Pareto-optimal solution of sub-MOP may not be the Pareto-optimal solution of the original MOP.

This theorem is one of the bases for MOEA/DVA. It seems that the condition in Theorem 4 is not easy to be satisfied. However, many existing benchmark test problems, such as continuous ZDT, DTLZ, UF1–UF10, WFG1, and WFG4–WFG5 problems, meet the condition of Theorem 4 as analyzed in Sections III-A and III-B. In Lemma 1 and Theorem 4, we just consider the independence of variables in the sub-MOP. Next, we consider the independence of subcomponents. Table V lists the differences between theorems and corollary in this section.

Theorem 5: If an MOP $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, $\mathbf{x} \in \Omega$ meets the following two conditions.

a) All decision variables $x_1, \ldots, x_n$ are distance variables and distance variables are divided into several subcomponents $\mathbf{x}_1, \ldots, \mathbf{x}_c$ by using Algorithm 3.
b) For arbitrary $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_c$, there exist $\bar{\mathbf{x}}_i, i = 1, \ldots, c$ such that $\mathbf{F}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \bar{\mathbf{x}}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_c) \leq \mathbf{F}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_c)$.

Then:

a) this MOP/sub-MOP only has one Pareto-optimal solution in the objective space;
b) all the subcomponents of this MOP/sub-MOP can be independently optimized one by one.

The proof of Theorems 5 can be found in Appendix B of supplementary material A.

Theorem 6: For an MOP $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, $\mathbf{x} \in \Omega$, let $\mathbf{x}_1$ be the diverse variables and $\mathbf{x}_2, \ldots, \mathbf{x}_c$ be the subcomponents of distance variables by using Algorithm 3. All diverse variables including mixed variables have an important effect on PF/PS if the following two conditions are met.

a) For arbitrary $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_c$, there exist $\bar{\mathbf{x}}_i, i = 2, \ldots, c$, such that $\mathbf{F}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \bar{\mathbf{x}}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_c) \leq \mathbf{F}(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_c)$.
b) The number of diverse variables (position variables and mixed variables) is equal to or less than the dimension of PF/PS.

The proof of Theorem 6 can be found in Appendix B of supplementary material A. If an MOP meets regularity property introduced in Section II-E, its dimension of PS is $(m-1)$. Continuous ZDT, DTLZ, and UF1–UF10 problems meet the two conditions of Theorem 6. When the number of position-related variables is set as $m-1$, WFG1–WFG7 problems also satisfy the two conditions of Theorem 6. For this kind of MOPs, mixed variables also have an important effect on PF/PS.

Corollary 1: For an MOP $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, $\mathbf{x} \in \Omega$, all diverse variables including mixed variables have an important effect on PF/PS if the following two conditions are met.

a) Distance variables are independent of one another for each objective function $f_i(\mathbf{x})$, $i = 1, \ldots, m$.
b) The number of diverse variables (position variables and mixed variables) is equal to or less than the dimension of PF/PS.

The proof of Corollary 1 can be found in Appendix B of supplementary material A. If an MOP meets regularity property introduced in Section II-E, its PS is $(m-1)$-dimension. To judge whether the mixed variables are important or not, Theorem 6 is suited to theoretical analysis while Corollary 1 is fit for computer implementation. Continuous ZDT, DTLZ, and UF1–UF10 problems meet two conditions of Corollary 1. When the number of position-related variables is set as $m-1$, WFG1, WFG4, and WFG5 problems satisfy two conditions of Corollary 1. Mixed variables also have an important effect on PF/PS for this kind of MOPs.

IV. EXPERIMENTAL STUDY

To show the performance of MOEA/DVA, the experimental studies are divided into six parts.

1) The first part is designed to show the effectiveness of the proposed algorithm by a comparison with current state-of-the-art MOEAs including nondominated sorting-based NSGA-III [4], hypervolume-based S metric selection-based evolutionary multiobjective optimization algorithms (SMS-EMOA) [3], and decomposition-based MOEA/D [2].

2) The second part is to show the effectiveness of the proposed algorithm on large-scale (in decision space) MOPs.
3) Taking NSGA-II [40] for example, the third part shows how to integrate the proposed subcomponent optimization into the framework of existing MOEAs.
4) The fourth part compares the proposed algorithm with other MOEAs based on linkage learning including multiple trajectory search (MTS) [41], regularity model-based multiobjective estimation of distribution algorithm (RMMEDA) [14], multiobjective iterated density–estimation evolutionary algorithm (MIDEA) [42], and multiobjective real-coded BOA (MrBOA) [43]. MTS combines random method with perturbation method to detect the interaction among decision variables. RMMEDA, MIDEA, and MrBOA use model building method (i.e., EDA) to learn the interaction among decision variables.
5) The fifth part compares the proposed algorithm with other MOEAs based on decomposition including MOEA based on decomposition (MOEA/D) [44], MOEA/D-M2M [36], and dynamical MOEA based on domain decomposition (DMOEADD) [45]. MOEA/DVA decomposes an MOP into a set of scalar subproblems by using position variable(s) and mixed variable(s). MOEA/D converts an MOP into a set of scalar subproblems by weight vectors. MOEA/D-M2M decomposes an MOP into a set of simple multiobjective optimization subproblems by using preference information (direction vectors). DMOEADD converts an MOP based on domain decomposition method.
6) The sixth part provides the sensitivity analysis for parameters NCA and NIA in analyzing and dividing decision variables. NCA represents the number of sampling solutions to recognize the control property of decision variable. NIA is the maximum number of tries required to judge the interaction between two variables.

The code of MOEA/D, MTS, and DMOEADD can be found in http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm. The code of RMMEDA can be found at: http://cswww.essex.ac.uk/staff/zhang/. The code of MOEA/D-M2M can be obtained from the authors. The code of SMS-EMOA can be found in Jmetal. The code of MrBOA can be found in the homepage of the original author. The codes of NSGA-III and MIDEA are, respectively, implemented following the suggestions of [4] and [42].

The complicated UF1–UF10 problems of CEC 2009 competition [13], ZDT4 [34], three-objective DTLZ1, DTLZ3 [35], and WFG4–WFG5 problems [12] are used. The number of decision variables and the maximum number of function evaluations are introduced in the corresponding experimental part. Other parameters of the compared algorithms are listed in Table A.1 of supplementary material A. All the compared algorithms quit when the function evaluation costs reach the maximum number. simulated binary crossover is used by NSGA-II, NSGA-III, SMS-EMOA, and MOEA/D on ZDT4, DTLZ1, DTLZ3, and WFG4–WFG5 problems. DE is used by MOEA/DVA, NSGA-III, SMS-EMOA, MOEA/D, MOEA/D-M2M, and DMOEADD on UF1–UF10 problems.

In the following experimental studies, the inverted generational distance (IGD) metric and $I_{\epsilon+}$ metric, which are comprehensive indexes of convergence and uniformity [46], are used. Let $\mathbf{P}^*$ be a number of uniformly distributed solutions on the PF. Suppose $\mathbf{P}$ is the approximate solution set of the PF, the average distance from $\mathbf{P}^*$ to $\mathbf{P}$ is defined as

$$\mathrm{IGD}\big(\mathbf{P}^*, \mathbf{P}\big) = \frac{\sum_{\mathbf{v}\in\mathbf{P}^*} d(\mathbf{v}, \mathbf{P})}{|\mathbf{P}^*|}$$

where $d(\mathbf{v}, \mathbf{P})$ is the minimal Euclidean distance from $\mathbf{v}$ to the solutions $\mathbf{P}$.

The additive ε-indicator ($I_{\varepsilon+}$) is defined as follows:

$$I_{\varepsilon+}\big(\mathbf{P}^*, \mathbf{P}\big) = \inf_{\varepsilon\geq 0}\Big\{\forall\ \mathbf{F}^2 \in \mathbf{P}^*,\ \exists\ \mathbf{F}^1 \in \mathbf{P} : \mathbf{F}^1 \prec_{\varepsilon+} \mathbf{F}^2\Big\}$$

where $\mathbf{F}^1 \prec_{\varepsilon+} \mathbf{F}^2$ means $\forall 1 \leq i \leq m, F_i^1 \leq \varepsilon + F_i^2$. Therefore, $I_{\varepsilon+}$-indicator gives an additional item by which each solution in $\mathbf{P}^*$ is dominated by the member of approximation solutions in $\mathbf{P}$.

To calculate the values of metrics, the number of solutions in set $\mathbf{P}^*$ is set as 500 for two-objective UF1–UF7 problems and 2500 for three-objective test problems. In the following experiments, each compared algorithm has been independently run 30 times to calculate the statistical values of metrics.

According to the average metric values, the value highlighted in bold in the experimental studies is the best result among the compared algorithms. Due to the randomness in MOEAs, statistical test procedures [47] are necessary to contrast the results obtained by optimization algorithms. In this paper, Wilcoxon's rank sum test at 0.05 is used to judge the significance of the differences between the solution set found by the best algorithm and each compared algorithm. According to Wilcoxon's rank sum test at 0.05 significance level, "+" means that the metric values of the best algorithm are significantly better than the compared algorithm.

A. MOEA/DVA Versus State-of-the-Art MOEAs on MOPs with Low-Dimensional Decision Variables

Low-dimensional decision variables here refer to decision variables whose values are less than or equal to 100. In this part of the experiments, UF1–UF10 problems with 30 variables and three-objective WFG4–WFG5 problems with 24 variables are used as the test problems. The maximum number of function evaluations is set as 300 000 for UF1–UF10 problems and 100 000 for WFG4–WFG5 problems. Their mathematical descriptions and the ideal PFs can be found in [12] and [44].

Table VI provides the average and standard deviation of $I_{\varepsilon+}$-metric and IGD-metric values of the final solutions obtained by MOEA/DVA, NSGA-III, SMS-EMOA, and MOEA/D. Table VII gives the statistical comparisons between the proposed MOEA/DVA and other compared algorithms. Table VIII shows the average CPU time spent by the compared algorithms on UF1–UF10 problem with 30 variables and 300 000 function evaluations. Fig. 12 illustrates the solution set with the median IGD values found by MOEA/DVA, NSGA-III, SMS-EMOA, and MOEA/D in the objective space.

TABLE VI

AVERAGE AND STANDARD DEVIATION OF $I_{\varepsilon+}$ METRIC AND IGD METRIC VALUES OBTAINED BY THE COMPARED ALGORITHMS ON UF1–UF10 PROBLEMS WITH 30 VARIABLES AND WFG4–WFG5 PROBLEMS WITH 24 VARIABLES. THE VALUE WITHIN PARENTHESES IS THE DEVIATION OF METRIC. THE VALUE IN BOLD IS THE BEST RESULT AMONG THE COMPARED ALGORITHMS BASED ON THE MEAN METRIC VALUE. ACCORDING TO WILCOXONS RANK SUM TEST AT 0.05, "+" IMPLIES THAT THE METRIC VALUES OF THE BEST ALGORITHM ARE SIGNIFICANTLY BETTER THAN THE COMPARED ALGORITHM

Metric	MOEA/DVA		NSGA-III		SMS-EMOA		MOEA/D	
	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$
UF1	**4.1350e-3**	**9.2148e-3**	6.5694e-3^{+}	1.6389e-2^{+}	1.2991e-2^{+}	5.1131e-2^{+}	5.0808e-3^{+}	1.5021e-2^{+}
	(9.9053e-5)	(9.6139e-4)	(3.9875e-3)	(1.3445e-3)	(3.9033e-3)	(2.1212e-2)	(1.0462e-1)	(1.0057e-2)
UF2	**4.1065e-3**	**9.0603e-3**	7.7898e-3^{+}	2.3086e-2^{+}	1.4459e-2^{+}	5.7067e-2^{+}	6.1240e-3^{+}	2.5338e-2^{+}
	(4.9092e-5)	(5.7421e-4)	(6.3919e-4)	(4.3280e-3)	(1.4518e-3)	(9.6093e-3)	(5.1602e-4)	(6.1677e-3)
UF3	2.2714e-2^{+}	5.6724e-2^{+}	2.9970e-2^{+}	8.7976e-2^{+}	3.4706e-2^{+}	9.6710e-2^{+}	**7.4994e-3**	**2.1551e-2**
	(7.2599e-3)	(1.3794e-2)	(2.7042e-2)	(4.0515e-2)	(2.4496e-2)	(5.6976e-2)	(3.6372e-3)	(1.1455e-2)
UF4	**3.5067e-2**	**4.3763e-2**	4.0436e-2^{+}	5.3668e-2^{+}	6.5663e-2^{+}	7.9705e-2^{+}	5.5163e-2^{+}	6.8570e-2^{+}
	(1.0070e-3)	(2.8428e-3)	(1.9958e-3)	(2.9918e-3)	(5.0141e-3)	(6.2694e-3)	(3.4085e-3)	(7.1297e-3)
UF5	**3.2592e-2**	**6.9379e-2**	3.9595e-2^{+}	9.4319e-2^{+}	5.0690e-1^{+}	6.3392e-1^{+}	1.8845e-1^{+}	3.1415e-1^{+}
	(4.6786e-3)	(1.5777e-2)	(3.4496e-2)	(5.8719e-2)	(1.5974e-1)	(1.4551e-1)	(7.9539e-2)	(1.3013e-1)
UF6	**5.6134e-2**	**9.3159e-2**	1.0795e-1^{+}	2.6174e-1^{+}	2.1106e-1^{+}	4.3104e-1^{+}	1.8611e-1^{+}	3.8414e-1^{+}
	(1.3729e-2)	(1.8142e-2)	(4.3731e-2)	(9.5953e-2)	(2.2583e-1)	(3.0544e-1)	(1.7664e-1)	(2.4237e-1)
UF7	**3.7667e-3**	**8.8102e-3**	7.8594e-3^{+}	3.8179e-2^{+}	8.4297e-3^{+}	5.8568e-2^{+}	5.0099e-3^{+}	2.2023e-2^{+}
	(4.6437e-5)	(1.4653e-3)	(2.3042e-3)	(1.6465e-2)	(1.2083e-3)	(9.0208e-3)	(3.3674e-4)	(6.7085e-3)
UF8	**5.7788e-2**	**1.4471e-1**	9.5301e-2^{+}	2.1646e-1^{+}	3.0704e-1^{+}	3.6727e-1^{+}	5.9993e-2^{+}	1.6221e-1^{+}
	(1.1960e-2)	(3.9164e-2)	(1.2106e-2)	(3.7338e-2)	(1.0690e-1)	(2.1164e-1)	(1.2990e-2)	(3.8858e-2)
UF9	1.2333e-1^{+}	1.8964e-1^{+}	**6.8933e-2**	**1.4862e-1**	2.0681e-1^{+}	3.3672e-1^{+}	1.3161e-1^{+}	3.0016e-1^{+}
	(1.6254e-1)	(2.0914e-1)	(8.3623e-3)	(3.4370e-2)	(1.0839e-1)	(1.5510e-1)	(7.1251e-2)	(1.7651e-1)
UF10	**1.0352e-1**	**2.0215e-1**	2.9143e-1^{+}	6.5321e-1^{+}	1.1115^{+}	1.1830^{+}	4.4116e-1^{+}	8.1656e-1^{+}
	(3.3009e-3)	(2.5121e-2)	(5.2710e-2)	(1.6599e-1)	(2.7426e-1)	(1.5065e-1)	(4.0000e-2)	(7.0977e-2)
WFG1	2.173^{+}	2.329^{+}	1.201^{+}	1.169^{+}	1.110^{+}	1.090^{+}	**2.054e-1**	**2.868e-1**
	(1.448e-2)	(1.595e-2)	(3.633e-3)	(1.572e-2)	(2.357e-2)	(2.309e-2)	(1.874e-2)	(2.904e-2)
WFG2	**2.220e-1**	**2.189e-1**	2.501e-1^{+}	2.996e-1^{+}	2.560e-1^{+}	2.360e-1^{+}	2.832e-2^{+}	2.557e-1^{+}
	(3.497e-2)	(9.726e-2)	(1.314e-2)	(2.984e-2)	(5.464e-3)	(7.744e-3)	(1.621e-2)	(3.416e-2)
WFG3	7.750e-2^{+}	1.887e-1^{+}	3.039e-1^{+}	3.578e-1^{+}	**7.254e-2**	**1.283e-1**	1.047e-1^{+}	2.051e-1^{+}
	(1.232e-2)	(2.849e-2)	(2.225e-2)	(3.873e-2)	(1.463e-2)	(2.453e-2)	(1.337e-2)	(3.419e-2)
WFG4	**2.2543e-1**	**2.3664e-1**	2.4235e-1^{+}	3.3052e-1^{+}	2.6131e-1^{+}	2.8076e-1^{+}	2.5522e-1^{+}	3.6883e-1^{+}
	(1.6949e-3)	(1.3056e-2)	(2.0532e-3)	(1.3117e-2)	(1.0094e-3)	(2.1231e-2)	(5.1022e-3)	(3.2294e-2)
WFG5	**2.1210e-1**	**2.9020e-1**	2.4051e-1^{+}	3.2724e-1^{+}	2.3727e-1^{+}	3.1025e-1^{+}	2.6102e-1^{+}	4.6008e-1^{+}
	(5.5937e-3)	(1.3595e-3)	(1.2191e-3)	(1.4327e-2)	(5.2118e-3)	(2.9456e-2)	(2.7675e-3)	(2.1571e-2)
WFG6	**2.190e-1**	**3.239e-1**	2.663e-1^{+}	4.073e-1^{+}	2.888e-1^{+}	3.884e-1^{+}	2.536e-1^{+}	3.426e-1^{+}
	(1.937e-3)	(2.621e-2)	(1.600e-2)	(3.848e-2)	(1.195e-2)	(4.305e-2)	(4.158e-3)	(9.170e-3)
WFG7	**2.150e-1**	**2.573e-1**	2.305e-1^{+}	3.469e-1^{+}	2.635e-1^{+}	2.792e-1^{+}	2.205e-1^{+}	3.466e-1^{+}
	(1.114e-3)	(2.815e-2)	(3.708e-3)	(2.776e-2)	(2.912e-3)	(2.107e-2)	(3.009e-3)	(4.686e-2)
WFG8	2.903e-1^{+}	5.217e-1^{+}	3.336e-2^{+}	4.647e-1^{+}	**2.584e-1**	**3.040e-1**	2.962e-1^{+}	5.375e-1^{+}
	(7.602e-3)	(1.032e-2)	(1.679e-2)	(4.325e-2)	(4.279e-3)	(2.770e-2)	(7.660e-3)	(4.931e-2)
WFG9	2.466e-1^{+}	4.559e-1^{+}	2.459e-1^{+}	4.214e-1^{+}	2.892e-1^{+}	5.077e-1^{+}	**2.390e-1**	**4.064e-1**
	(1.896e-2)	(7.192e-2)	(1.064e-2)	(5.471e-2)	(2.078e-3)	(1.729e-2)	(9.982e-3)	(8.948e-2)

TABLE VII

STATISTICS OF PERFORMANCE COMPARISONS BETWEEN MOEA/DVA AND OTHER COMPARED MOEAS. ACCORDING TO WILCOXON RANK SUM TEST AT 0.05 SIGNIFICANCE LEVEL, "+," "−," AND "≈," RESPECTIVELY, ARE THE NUMBER OF MOEA/DVA THAT ARE BETTER THAN, WORSE THAN, AND SIMILAR TO OTHER COMPARED ALGORITHMS

MOEA/DVA VS.	T test	NSGA-III	SMS-EMOA	MOEA/D	MTS	RMMEDA	MrBOA	MOEA/D-M2M	DMOEADD	MIEDA
IGD	+	16	16	15	8	10	10	9	10	10
	-	3	3	3	2	0	0	1	0	0
	≈	0	0	1	0	0	0	0	0	0
$I_{\varepsilon+}$	+	15	16	15	8	10	10	9	10	10
	-	3	3	3	2	0	0	1	0	0
	≈	1	0	1	0	0	0	0	0	0

In terms of average $I_{\varepsilon+}$-metric and IGD-metric, Table VI shows that the proposed MOEA/DVA is the best on UF1–UF2, UF4–UF8, UF10, WFG2, and WFG4–WFG7 problems. NSGA-III performs best on UF9 problem. SMS-EMOA does the best on WFG3 and WFG8 problems, while MOEA/D performs the best on UF3, WFG1, and WFG9 problems. MOEA/DVA outperforms NSGA-III in 16 out of 19 comparisons based on $I_{\varepsilon+}$ metric and IGD metric as shown in Table VII. According to Wilcoxon's rank sum test at 0.05 significance level, MOEA/DVA significantly outperforms SMS-EMOA in 16 out of 19 comparisons. Moreover, MOEA/DVA significantly outperforms MOEA/D in 15 out of 19 comparisons based on $I_{\varepsilon+}$ and IGD metrics.

As shown in Fig. 12, MOEA/DVA performs the best in the diversity and approximation quality of found solutions on UF1–UF2 and UF4–UF10 problems, while MOEA/D is the best in the uniformity and convergence of obtained solutions on UF3 problem. The success of MOEA/DVA comes

TABLE VIII
AVERAGE CPU TIME (IN SECONDS) SPENT BY THE COMPARED ALGORITHMS ON UF1–UF10 PROBLEM WITH 30 VARIABLES AND 300 000 FUNCTION EVALUATIONS

Problem	MOEA/DVA	NSGA-III	SMS-EMOA	MOEA/D	MTS	RMMEDA	MrBOA	MOEA/D-M2M	DMOEADD	MIEDA
UF1	0.788+	27.59+	85.25+	7.576+	**0.689**	150.1+	40.12+	81.24+	78.62+	8.793+
UF2	**1.005**	27.82+	185.6+	7.824+	1.187	144.8+	38.91+	79.56+	80.68+	8.621+
UF3	**1.388**	28.09+	230.4+	8.562+	1.561+	150.9+	31.84+	90.43+	87.67+	8.950+
UF4	1.017+	27.46+	161.7+	8.240+	**0.875**	142.7+	22.76+	76.45+	79.38+	6.729+
UF5	**0.723**	27.93+	47.98+	7.935+	0.892+	142.1+	42.41+	74.25+	76.55+	9.986+
UF6	1.252+	28.15+	49.83+	8.037+	**0.983**	148.5+	35.86+	88.49+	75.43+	10.37+
UF7	0.799+	28.57+	525.0+	7.673+	**0.522**	148.6+	36.36+	72.94+	78.91+	9.297+
UF8	**0.847**	81.65+	4806+	14.53+	1.719+	160.3+	29.52+	84.27+	105.5+	11.54+
UF9	**0.803**	75.36+	1829+	13.82+	1.219+	158.8+	34.43+	82.76+	107.5+	12.12+
UF10	1.171+	80..92+	105.6+	14.49+	**0.891**	163.9+	38.57+	83.46+	101.3+	11.65+

from two aspects. One is that MOEA/DVA has a faster convergence ability than the three current state-of-the-art MOEAs including NSGA-III, SMS-EMOA, and MOEA/D. The proposed algorithm reduces the optimization hardness of MOP by decomposing distance variables into a set of smaller components. Each individual/sub-MOP independently optimizes components one by one to speed up the convergence of the population in the early stages of evolution. The other is that MOEA/DVA evolves all decision variables to optimize the uniformity of population in the objective space in the late stages of evolution. Optimizing the convergence of population first and then optimizing the uniformity of population may be the reason for the success of MOEA/DVA. The proposed subcomponent optimization makes the main contribution to the performance of MOEA/DVA. NSGA-III, SMS-EMOA, and MOEA/D treat all decision variables as a whole to optimize. If the MOP to be solved is decomposable, it is reasonable to discover the problem structure and use it to reduce the optimization difficulty of this MOP. The proposed algorithm MOEA/DVA is based on this idea.

For most UF1–UF10 problems, we want to explain why our proposed algorithm provides better results than the other compared algorithms. The reason is that these MOPs satisfy the following condition. All the distance variables are independent of one another for each objective function as shown in Figs. 5 and 6. According to Theorem 4, all the distance variables of each sub-MOP can be independently optimized one by one. Roughly speaking, MOEA/DVA divides the search in n-dimensional decision space into two parts.

1) $N * (n - m + 1)$ 1-D search. There are N sub-MOPs/individuals needing to be optimized and each sub-MOP/individual has $(n - m + 1)$ subcomponents $\{x_m\}, \dots, \{x_n\}$ requiring to be optimized.
2) $(m-1)$-dimensional uniformity optimization for diverse variables $x_1, \dots, x_{m-1}$. On the contrary, NSGA-III, SMS-EMOA, and MOEA/D need to search in n-dimensional space. N is the number of sub-MOPs or size of population, n is the number of decision variables, and m is the number of objective functions. Why does MOEA/DVA reduce the dimensions of the search space? The reason is that MOEA/DVA learns the variable linkages and divides the distance variables with high dimension into several independent low-dimensional subcomponents to optimize.

MOEA/DVA does not work well on WFG problems. The reason may come from the following aspects.

1) Most WFG problems are not difficult MOPs. NSGA-III, SMS-EMOA, and MOEA/D can converge rapidly at 20 000 to 40 000 function evaluations as shown in supplementary material B. Therefore, the problems are not difficult enough to show the effectiveness of subcomponent optimization of MOEA/DVA.
2) For some WFG problems, the mapping of MOP from PS to PF is highly biased and MOEA/DVA does not have enough function evaluations to perform uniformity optimization. An example is WFG1 problem as shown supplementary material B.

The proposed MOEA/DVA has dependency analysis for decision variables, while NSGA-III, SMS-EMOA, and MOEA/D have no dependency analysis. The four compared algorithms use the same DE operator. Therefore, this part of the experiments also want to demonstrate the effectiveness of subcomponent optimization/dependency analysis in MOEA/DVA to some extent.

As shown in Tables VIII and IX, the average CPU time (in seconds) spent by the proposed algorithm is greatly less than NSGA-III, SMS-EMOA, and MOEA/D. The reason is analyzed in Table X. The most important reason is the time used for one child to update the evolutionary population. As shown in step 3 of Algorithm 5, MOEA/DVA spends $O(m)$ time complexity in the early stages of evolution by using a child to update its parent. Using one child to update the evolutionary population, NSGA-III needs $O(mN)$ time complexity, MOEA/D spends $O(mT), T = 0.1N$ time complexity, and SMS-EMOA spends $O(N \log(N))$ time complexity for two-objective MOPs and $O(N^3)$ time complexity for three-objective MOPs.

By doing this part of the experiments, one of the aims is to show that the first advantage of our proposed algorithm is its low computational cost.

B. MOEA/DVA Versus State-of-the-Art MOEAs on Large-Scale MOPs

Large-scale MOPs here refer to MOPs with more than 100 decision variables. The existing MOEAs [2]–[4] tend to validate their effectiveness on MOPs with low-dimensional decision variables. In fact, many real-world problems have

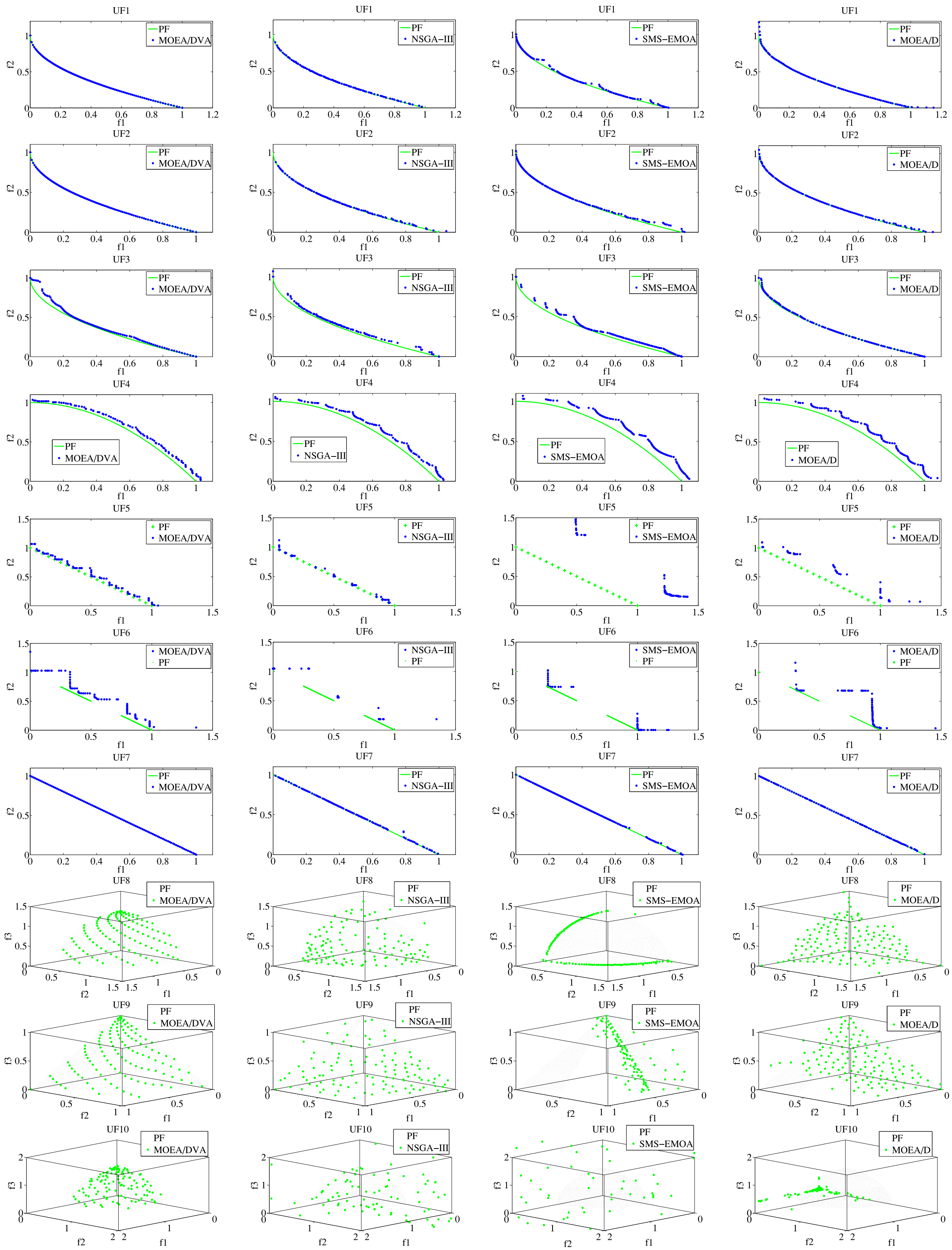


Fig. 12. Solution set with the median IGD values found by MOEA/DVA, NSGA-III, SMS-EMOA, and MOEA/D on UF1–UF10 problems with 30 variables and 300 000 function evaluations.

hundreds or even thousands of decision variables. Therefore, we want to study the effectiveness of the proposed subcomponent optimization/dependence analysis on large-scale MOPs.

In this part of the experiments, the number of decision variables for all MOPs is set as 200. The maximum number of function evaluations is set as 1 200 000 for ZDT4, DTLZ1,

TABLE IX

STATISTICS OF CPU TIME COMPARISONS BETWEEN MOEA/DVA AND OTHER COMPARED MOEAS. ACCORDING TO WILCOXON RANK SUM TEST AT 0.05 SIGNIFICANCE LEVEL, "+," "−," AND "≈," RESPECTIVELY, ARE THE NUMBER OF MOEA/DVA THAT ARE BETTER THAN, WORSE THAN, AND SIMILAR TO OTHER COMPARED ALGORITHMS

MOEA/DVA VS.	T test	NSGA-III	SMS-EMOA	MOEA/D	MTS	RMMEDA	MrBOA	MOEA/D-M2M	DMOEADD	MIEDA
CPU time	+	10	10	10	4	10	10	10	10	10
	-	0	0	0	5	0	0	0	0	0
	≈	0	0	0	1	0	0	0	0	0

TABLE X

DIFFERENCES IN TIME COMPLEXITY AMONG THE COMPARED ALGORITHM. m IS THE NUMBER OF OBJECTIVE FUNCTIONS. n IS THE NUMBER OF DECISION VARIABLES. N IS THE POPULATION SIZE

Time complexity	MOEA/DVA	NSGA-III	SMS-EMOA	MOEA/D
Use one offspring to update the evolutionary population	Early: $O(m)$ Late: $O(mT), T = 0.1N$	$O(mN)$	2-objective: $O(N \log(N))$ 3-objective: $O(N^3)$	$O(mT), T = 0.1N$

TABLE XI

MEAN AND STANDARD DEVIATION OF IGD METRIC AND $I_{\varepsilon+}$ METRIC VALUES OBTAINED BY THE COMPARED ALGORITHMS ON MOPS WITH 200 VARIABLES. THE VALUE WITHIN PARENTHESES INDICATES THE DEVIATION OF METRIC

Metric	MOEA/DVA		NSGA-III		SMS-EMOA		MOEA/D	
	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$
ZDT4(200)	**3.9231e-3**	**7.2229e-3**	2.8054e-2^{+}	2.9220e-2^{+}	1.7910e+2^{+}	1.8064e+2^{+}	8.4503e-1^{+}	1.0060^{+}
	(3.5337e-5)	(2.5025e-6)	(4.0329e-3)	(3.0814e-3)	(4.7488e+1)	(5.2619e+1)	(1.5516e-3)	(1.8674e-3)
DTLZ1(3,200)	**2.2652e-2**	**3.8624e-2**	6.0585e+1^{+}	3.6903e+1^{+}	3.6777e+1^{+}	2.4997e+1^{+}	3.5401e+2^{+}	2.2919e+2^{+}
	(1.1502e-4)	(5.2420e-4)	(7.6095)	(4.7209)	(9.4812e+1)	(6.4762e+1)	(3.1674e+1)	(2.0112e+1)
DTLZ3(3,200)	**5.8425e-2**	**1.4525e-1**	1.1319e+2^{+}	7.0170e+1^{+}	8.3956e+1^{+}	5.8058e+1^{+}	9.7147e+2^{+}	6.4828e+2^{+}
	(1.7293e-4)	(2.0024e-4)	(8.1592)	(4.9003)	(2.0722e+2)	(1.4575e+2)	(1.6708e+2)	(1.0911e+2)
UF1(200)	**4.0108e-3**	**8.6185e-3**	6.0808e-2^{+}	1.3290e-1^{+}	5.7179e-2^{+}	1.5141e-1^{+}	5.8359e-2^{+}	1.5199e-1^{+}
	(8.1582e-5)	(2.5630e-4)	(1.1413e-2)	(2.4411e-2)	(9.2118e-3)	(1.4365e-2)	(1.0714e-2)	(2.8984e-2)
UF2(200)	**4.0657e-3**	**8.5908e-3**	4.2506e-2^{+}	1.0282e-1^{+}	4.3679e-2^{+}	1.1229e-1^{+}	4.7182e-2^{+}	1.6872e-1^{+}
	(2.1424e-5)	(3.7614e-4)	(1.8169e-3)	(5.2937e-3)	(2.9118e-3)	(1.2244e-2)	(1.4869e-2)	(5.0220e-2)
UF3(200)	**3.9059e-3**	**8.5995e-3**	1.1255e-2^{+}	1.8061e-2^{+}	6.7450e-3^{+}	1.1545e-2^{+}	1.8507e-2^{+}	4.6590e-2^{+}
	(5.0902e-5)	(7.9138e-4)	(6.3752e-4)	(1.9780e-3)	(6.5540e-4)	(1.8072e-3)	(7.4514e-3)	(1.9008e-2)
UF4(200)	**3.2392e-2**	**3.4079e-2**	1.0993e-1^{+}	1.0084e-1^{+}	1.1833e-1^{+}	1.0314e-1^{+}	1.1859e-1^{+}	1.1105e-1^{+}
	(2.8345e-4)	(6.7679e-4)	(1.0199e-2)	(5.1304e-3)	(4.8842e-3)	(2.7272e-3)	(3.8865e-3)	(1.7441e-3)
UF5(200)	**3.2378e-2**	**5.8719e-2**	4.1774e-1^{+}	5.3229e-1^{+}	1.8144^{+}	1.5251^{+}	1.8667e-1^{+}	3.8453e-1^{+}
	(5.2747e-3)	(6.0963e-3)	(9.2097e-2)	(1.4783e-1)	(1.5967e-1)	(8.6633e-2)	(3.5638e-2)	(1.2395e-1)
UF6(200)	**1.8064e-2**	**2.8861e-2**	1.9761e-1^{+}	4.0404e-1^{+}	2.4355e-1^{+}	4.3114e-1^{+}	5.9728e-2^{+}	1.5387e-1^{+}
	(2.5301e-3)	(3.6456e-3)	(6.2168e-2)	(1.2695e-1)	(1.5323e-1)	(1.8339e-1)	(3.6105e-2)	(6.4987e-2)
UF10(200)	**2.3715e-1**	**2.8664e-1**	1.5102^{+}	1.4776^{+}	1.3859^{+}	1.6517^{+}	1.4677^{+}	1.3188^{+}
	(4.8283e-1)	(3.9242e-1)	(3.3453e-1)	(2.1995e-1)	(3.3492e-1)	(2.1862e-1)	(2.6402e-1)	(1.7627e-1)

DTLZ3, and UF1–UF2 problems and 3 000 000 for UF3–UF6 and UF10 problems. Their mathematical descriptions and the ideal PFs can be found in [34], [35], and [44].

Table XI gives the average and standard deviation of $I_{\varepsilon+}$-metric and IGD-metric values of the obtained solutions by MOEA/DVA, NSGA-III, SMS-EMOA, and MOEA/D. In terms of average $I_{\varepsilon+}$-metric and IGD-metric, we can see that MOEA/DVA is better than NSGA-III, SMS-EMOA, and MOEA/D on the selected MOPs with 200 variables. According to Wilcoxon's rank sum test at 0.05 level, MOEA/DVA significantly outperforms the other three compared algorithms in all of comparisons based on $I_{\varepsilon+}$-metric and IGD-metric values on the selected large-scale MOPs.

Fig. 13 shows the evolution process of the mean of IGD-metric values by the four compared algorithms on ZDT4, DTLZ1, DTLZ3, UF1–UF6, and UF10 problems with 200 variables. We can see that MOEA/DVA converges faster and performs better than NSGA-III, SMS-EMOA, and MOEA/D on all selected MOPs with 200 variables. The reason for the success of the proposed algorithm may lie in decomposing the distance variables into a set of smaller subcomponents. Each individual/sub-MOP independently optimizes subcomponents one by one. On the contrary, NSGA-III, SMS-EMOA, and MOEA/D treat all decision variables as a whole to optimize. For large-scale complex MOPs, treating all decision variables as a whole to optimize may make these algorithms converge prematurely as shown in Fig. 13.

Through this part of the experiments, another advantage of our proposed algorithm is that it has good scalability to MOPs with large number of decision variables.

C. Integrating Proposed Subcomponent Optimization Into Existing MOEAs

In this part, we take NSGA-II [40] for example to show how to integrate the proposed subcomponent optimization into the framework of existing MOEAs. That is to say, we want to investigate what benefits can be obtained by integrating the proposed mechanism. Algorithm 7 gives a brief description of NSGA-II and Algorithm 8 provides the detail of how to integrate the proposed mechanism into NSGA-II.

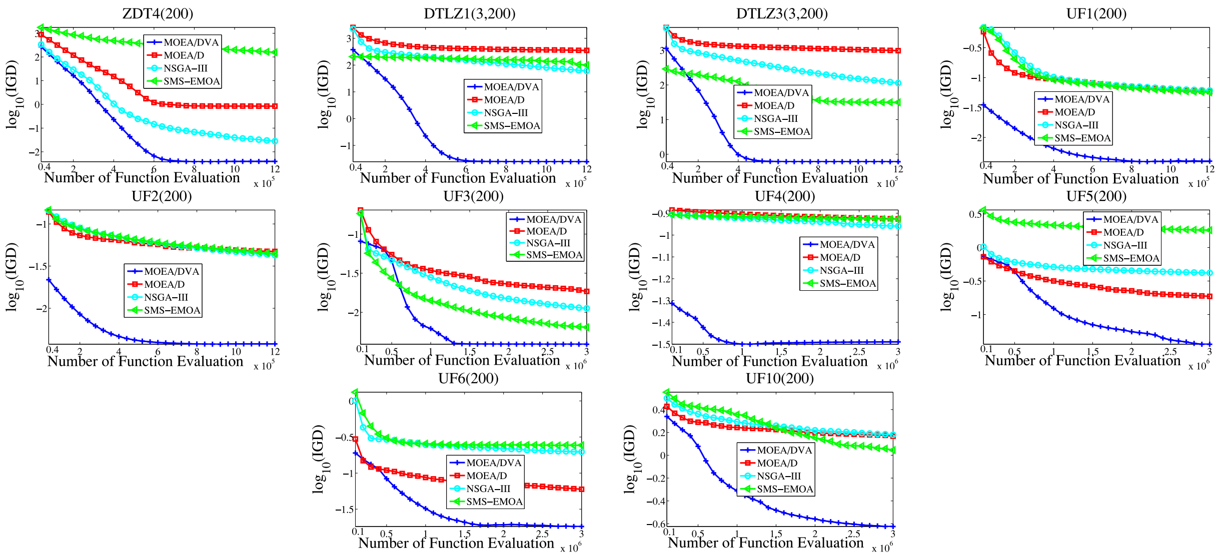


Fig. 13. Plot the evolution process of the mean of IGD-metric values by MOEA/DVA, NSGA-III, SMS-EMOA, and MOEA/D on MOPs with 200 variables.

Algorithm 7 Brief Description of NSGA-II [40]

Require: N: the population size.

1. Initialize evolutionary population randomly.
2. Evolution:
   - 2.1 generate offspring population
     - For $i = 1 : N$
       - Use evolutionary operators (crossover and mutation) to generate an offspring and evaluate it.
     - End for
   - 2.2 Use nondominated sorting and crowd distance to select the next parent population.
   - 2.3 If the stopping criterion is met, stop; else go to Step 2.

Algorithm 8 NSGA-II-DVA

Require: N: the population size.

1. Use lines 1-5 of Algorithm 4 to initialize and evolve the population.
2. Evolution:
   - 2.1 generate offspring population
     - For $i = 1 : N$
       - If *rand* < 0.95
         - Use evolutionary operators (crossover and mutation) to generate an offspring and evaluate it.
       - Else
         - Use lines 9-12 of Algorithm 4 to evolve i-th individual of evolutionary population.
       - End If
     - End For
   - 2.2 Use nondominated sorting and crowd distance to select the next parent population.
   - 2.3 If the stopping criterion is met, stop; else go to Step 2.

The differences between NSGA-II and NSGA-II-DVA are the initialization of population and evolutionary operators. NSGA-II-DVA initializes the evolutionary population, learns the control property of variable and linkage between two decision variables, and divides the distance variables. Then NSGA-II-DVA uses the evolutionary operators in the original NSGA-II as global search operators and subcomponent optimization as local search operator to evolve the population. If the ith individual is selected to evolve, NSGA-II-DVA optimizes its subcomponents one by one for this individual. In Algorithm 8, rand is a uniform random number in [0, 1).

In this part, DTLZ1, DTLZ3, UF1, UF3, UF5–UF6, UF8, and UF10 problems with 200 variables are used as the test problems. The maximum number of function evaluations is set as 1 200 000 for all selected MOPs.

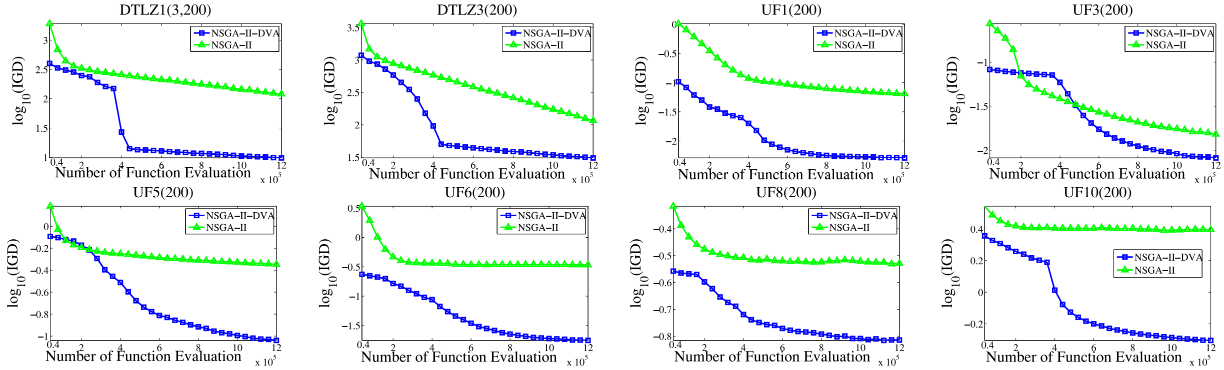


Fig. 14. Evolution process of the average of IGD-metric values by NSGA-II-DVA and NSGA-II on MOPs with 200 variables and 1 200 000 function evaluations.

TABLE XII
MEAN AND STANDARD DEVIATION OF $I_{\varepsilon+}$ METRIC AND IGD METRIC VALUES FOUND BY NSGA-II-DVA AND NSGA-II ON MOPs WITH 200 VARIABLES AND 1 200 000 FUNCTION EVALUATIONS. THE VALUE WITHIN PARENTHESES IS THE DEVIATION OF METRIC

Metric	NSGA-II-DVA		NSGA-II	
	IGD	$I_{\varepsilon+}$	IGD	$I_{\varepsilon+}$
DTLZ1(3,200)	**9.7738** (1.4787)	**6.3100** (1.1278)	1.2012e+2+ (8.3658)	7.9127e+1+ (8.7025)
DTLZ3(3,200)	**3.0379e+1** (8.5190)	**1.9769e+1** (5.5118)	1.1609e+2+ (1.4069e+1)	7.3507e+1+ (8.6760)
UF1(200)	**5.1240e-3** (1.0548e-4)	**1.1509e-2** (1.1772e-3)	6.4526e-2+ (1.2217e-2)	1.4259e-1+ (2.1355e-2)
UF3(200)	**8.1762e-3** (4.9351e-4)	**1.5887e-2** (1.2775e-3)	1.5228e-2+ (3.2702e-3)	2.3291e-2+ (4.7971e-3)
UF5(200)	**9.1705e-2** (4.7886e-2)	**1.5948e-1** (6.8345e-2)	4.5016e-1+ (1.0345e-1)	6.1727e-1+ (1.3645e-1)
UF6(200)	**1.7787e-2** (3.7780e-3)	**6.2652e-2** (1.1661e-2)	3.4356e-1+ (1.0546e-1)	5.0585e-1+ (1.2295e-1)
UF8(200)	**1.5406e-1** (6.1707e-2)	**2.7075e-1** (1.0018e-1)	2.9694e-1+ (1.0673e-2)	7.3518e-1+ (5.7290e-3)
UF10(200)	**4.9777e-1** (4.0446e-1)	**7.0108e-1** (3.0794e-1)	2.4819+ (2.0319e-1)	1.9261+ (1.1825e-1)

Table XII presents the average and standard deviation of the $I_{\varepsilon+}$ metric and IGD-metric values of the final solutions obtained by NSGA-II and NSGA-II-DVA. According to Wilcoxon's rank sum test at 0.05 significance level, NSGA-II-DVA significantly outperforms NSGA-II on all of the selected MOPs with 200 variables based on IGD-metric and $I_{\varepsilon+}$-metric values.

Fig. 14 shows the evolution process of the average of IGD-metric values by NSGA-II and NSGA-II-DVA on DTLZ1, DTLZ3, UF1, UF3, UF5–UF6, UF8, and UF10 problems with 200 variables and 1 200 000 function evaluations. By integrating the subcomponent optimization, NSGA-II-DVA converges faster and performs better than NSGA-II on the selected MOPs.

Recently, Durillo *et al.* [48] suggested that performing a fixed number of function evaluations may not offer information about the effort needed by an algorithm to obtain satisfactory solutions. In supplementary material C, the effect of parameter scalability by MOEA/DVA, NSGA-III, SMS-EMOA, and MOEA/D is studied. The behaviors of the four compared algorithms on UF1–UF7 problems with 10, 30, 100, 300, and 1000 variables are investigated. By this way, we want to study which algorithm behaves more efficiently when solving problems with an increasing number of variables.

From these experiments, it can be discovered that our proposed algorithm has its third advantage: portability. By doing this part of the experiments, we also want to show the effectiveness of the proposed subcomponent optimization/linkage learning.

D. MOEA/DVA Versus Other MOEAs Based on Linkage Learning

In order to study the effectiveness of the linkage learning method used in this paper, MOEA/DVA is compared with MTS, MrBOA, MIDEA, and RMMEDA. This part of the experiments can be found in supplementary material D.

E. MOEA/DVA Versus Other MOEAs Based on Decomposition

To observe the effectiveness of the proposed MOP decomposition based on diverse variables, MOEA/DVA is compared with MOEA/D, MOEA/D-M2M, and DMOEADD. This part of the experiments can be found in supplementary material D.

F. Sensitivity Analysis of Parameters

There are two control parameters in the division of distance variables in MOEA/DVA.

1) NCA represents the number of sampled solutions to recognize the control property of decision variables. NCA affects the precision of control analysis of decision variables.
2) NIA is the maximum number of tries required to judge the interaction between two variables. NIA determines the precision of learned variable linkages.

To study how MOEA/DVA is sensitive to the above two parameters, the experimental results can be found in supplementary material D.

V. CONCLUSION

Decomposition of decision variables is popular in cooperative coevolution for single objective optimization. However, there is little work in introducing variable dividing techniques to help solve multiobjective optimization problems. In contrast, most MOEAs treat all the decision variables as a whole to optimize. These algorithms may not be very good for difficult MOPs.

This paper proposes a simple evolutionary multiobjective optimization algorithm based on DVAs, named as MOEA/DVA. The DVAs include control property analysis and variable linkage analysis. Based on diverse variables (position variables and mixed variables), MOEA/DVA decomposes a complicated MOP into a set of simpler sub-MOPs. The distance variables are divided into several low-dimensional subcomponents based on learned variable linkages. Each sub-MOP independently optimizes subcomponents one by one.

In the experiment studies, three advantages of the proposed algorithm are investigated: 1) low computation cost; 2) scalability to large number of decision variables; and 3) portability. Experimental results show the effectiveness of subcomponent optimization/linkage learning. MOEA/DVA has also been compared with MTS, MrBOA, and RMMEDA on UF1–UF10 problems. In order to show the effectiveness of the proposed decomposition MOP based on diverse variables, MOEA/DVA has been compared with MOEA/D, MOEA/D-M2M, and DMOEADD on UF1–UF10 problems. Our analysis has shown that MOEA/DVA obtains better convergence than the three compared algorithms. However, the distribution of the solutions found by MOEA/DVA is dependent on the mapping of the problem from PS to PF to some extent.

Future work includes the following aspects.

1) In this paper, we learn the variable linkages at the beginning stage of evolution. The linkages learned should be accurate because wrongly identified subcomponents make the convergence difficult. It may be more intelligent to learn the variable linkages in the process of evolution [49], [50].
2) To separate the optimization difficulty of MOP and the conflict among objective functions, MOEA/DVA fixes the values of the diverse variables of the population at the stage of subcomponent optimization. From the experimental study, fixing the values of the diverse variables of the population has a defect. That is to say, the distribution of solutions found by MOEA/DVA is dependent on the mapping from PS of MOP to its PF. How to deal with diverse variables, including mixed variables and position variables, is an open problem and an interesting direction.
3) The mixed variables are currently treated as position variables. However, a mixed variable affects not only the spread of generated solutions but also the convergence of generated solutions. How to deal with mixed variables better is also an open problem. To analyze the dynamic features of mixed variables in the process of evolution may be an interesting direction.
4) For simplicity, all subcomponents and sub-MOPs are respectively treated equally in this paper and the same computational resource has been assigned to each subcomponent and sub-MOP, respectively. However, the number of decision variables in each subcomponent may be different and each subproblem may be at a different stage of evolution and has different computational difficulties. Therefore, it is reasonable to assign different computational resource to different subcomponents [38] and sub-MOPs based on their recent performance.
5) The exclusive variable is another feature of decision variables. The general definition of an exclusive variable [51] can be stated as follows. If x_j just has an effect on $f_i(\mathbf{x})$ but no effect on the other objective functions, then x_j is called an exclusive variable of $f_i(\mathbf{x})$. All the exclusive variables of $f_i(\mathbf{x})$ are recorded as $\mathbf{x}_i^{\text{Exc}}$, while the decision variables related to multiple objective functions are defined as: $\mathbf{x}^{\text{Common}} = \{x_1, \ldots, x_n\} \setminus \cup_{i=1}^{m} \mathbf{x}_i^{\text{Exc}}$. Changing decision variables of $\mathbf{x}_i^{\text{Exc}}$ in $\mathbf{x} = (x_1, \ldots, x_n)$ never causes incomparable decision vectors because all the decision variables in $\mathbf{x}_i^{\text{Exc}}$ can only affect one objective function. Therefore, the decision variables in $\cup_{i=1}^{m} \mathbf{x}_i^{\text{Exc}}$ only have an effect on the convergence of generated solutions. Furthermore, the decision variable in $\mathbf{x}^{\text{Common}}$ is related to the convergence or/and diversity of generated solutions.
6) Goh and Tan [52] used competitive-cooperative coevolutionary paradigm to deal with MOPs. This paper has two significant mechanisms.
   a) The ith subpopulation is used to evolve the ith variable. Each subpopulation competes to stand for a subcomponent of the MOP.
   b) The eventual winners cooperate to generate better solutions. It is a good idea to use multiple populations and competition mechanism to learn the subcomponent.
7) Most preference information [32], such as reference point, preference direction, and value function, comes from the objective space. Little work has reported the preference information based on the feature of decision variables. Qi *et al.* [53] offered an instance of MOP with preference information coming from the decision space.
8) Other interesting issues include analysis of fitness function [54], parameter control [55], locating multiple optimal solutions of nonlinear equation systems [56], using visualization of Pareto-front approximations [57], decomposition [58], two-archive algorithm [59], and reduced objective computations [60] for many objective problems.

Acknowledgment

The authors would like to thank Q. Zhang, K. Tang, Z. Yang, and H. Wang, who have helped them to turn an idea into this paper.

References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[2] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[3] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[4] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[5] T. Weise, R. Chiong, and K. Tang, "Evolutionary optimization: Pitfalls and booby traps," *J. Comput. Sci. Technol.*, vol. 27, no. 5, pp. 907–936, 2012.

[6] M. Potter and K. Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Int. Conf. Parallel Probl. Solv. Nat.*, vol. 2. Jerusalem, Israel, 1994, pp. 249–257.
[7] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.
[8] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.
[9] Y. Mei, X. Li, and X. Yao, "Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 435–449, Jun. 2014.
[10] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[11] Y. Chen, T. Yu, K. Sastry, and D. Goldberg, "A survey of linkage learning techniques in genetic and evolutionary algorithms," Illinois Genet. Algorithms Libr., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, Tech. Rep. 2007014, 2007.
[12] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
[13] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electr. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-887, 2008.
[14] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.
[15] L. Jiao, Y. Li, M. Gong, and X. Zhang, "Quantum-inspired immune clonal algorithm for global optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1234–1253, Oct. 2008.
[16] K. Tang, X. Li, P. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization," Nat. Inspir. Comput. Appl. Lab., Univ. Sci. Technol. China, Hefei, China, Tech. Rep. 2010001, 2010.
[17] T. Yu, D. Goldberg, K. Sastry, C. Lima, and M. Pelikan, "Dependency structure matrix, genetic algorithms, and effective recombination," *Evol. Comput.*, vol. 17, no. 4, pp. 595–626, 2009.
[18] W. Chen, T. Weise, Z. Yang, and K. Tang, "Large-scale global optimization using cooperative coevolution with variable interaction learning," in *Proc. Conf. Parallel Probl. Solv. Nat.*, Kraków, Poland, 2010, pp. 300–309.
[19] P. Toint, "Test problems for partially separable optimization and results for the routine PSPMIN," Dept. Math., Univ. Namur, Namur, Belgium, Tech. Rep. 83/4, 1983.
[20] B. Colson and P. Toint, "Optimizing partially separable functions without derivatives," *Optim. Methods Softw.*, vol. 20, nos. 4–5, pp. 493–508, 2005.
[21] M. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 378–393, Jun. 2014.
[22] D. Thierens and D. Goldberg, "Mixing in genetic algorithms," in *Proc. 5th Int. Conf. Genet. Algorithms*, Urbana, IL, USA, 1993, pp. 38–45.
[23] M. Munetomo and D. Goldberg, "Identifying linkage groups by nonlinearity/nonmonotonicity detection," in *Proc. Genet. Evol. Comput. Conf.*, vol. 1. Orlando, FL, USA, 1999, pp. 433–440.
[24] M. Tezuka, M. Munetomo, and K. Akama, "Linkage identification by nonlinearity check for real-coded genetic algorithms," in *Proc. Genet. Evol. Comput. Conf.*, Seattle, WA, USA, 2004, pp. 222–233.
[25] K. Weicker and N. Weicker, "On the improvement of coevolutionary optimizers by learning variable interdependencies," in *Proc. IEEE Congr. Evol. Comput.*, Washington, DC, USA, 1999, pp. 1627–1632.
[26] Y. Chen, "Extending the scalability of linkage learning genetic algorithms: Theory and practice," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Urbana, IL, USA, 2004.
[27] J. E. Smith, "Self adaptation in evolutionary algorithms," Ph.D. dissertation, Dept. Comput. Sci., Univ. West England, Bristol, U.K., 1998.
[28] Q. Zhang and H. Muehlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 127–136, Apr. 2004.
[29] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287–297, Nov. 1999.
[30] M. Pelikan and D. Goldberg, "BOA: The Bayesian optimization algorithm," in *Proc. Genet. Evol. Comput. Conf.*, Orlando, FL, USA, 1999, pp. 525–532.
[31] T. Yu, A. Yassine, and D. Goldberg, "A genetic algorithm for developing modular product architectures," in *Proc. ASME Int. Design Eng. Tech. Conf.*, Chicago, IL, USA, 2003, pp. 515–524.
[32] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer Academic, 1999.
[33] C. Hillermeier, *Nonlinear Multiobjective Optimization—A Generalized Homotopy Approach*. Boston, MA, USA: Birkhauser, 2001.
[34] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
[35] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput.*, Honolulu, HI, USA, 2002, pp. 825–830.
[36] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.
[37] K. Fang and D. Lin, "Uniform designs and their application in industry," in *Handbook on Statistics in Industry*, vol. 22. Amsterdam, The Netherlands: Elsevier, 2003, pp. 131–170.
[38] M. Omidvar, X. Li, and X. Yao, "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms," in *Proc. Genet. Evol. Comput. Conf.*, Dublin, Ireland, 2011, pp. 1115–1122.
[39] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[40] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[41] L. Tseng and C. Chen, "Multiple trajectory search for unconstrained/constrained multi-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 1951–1958.
[42] P. Bosman and D. Thierens, "The naive MIDEA: A baseline multi-objective EA," in *Proc. Evol. Multi-Criterion Optim. (EMO)*, Guanajuato, Mexico, 2005, pp. 428–442.
[43] C. Ahn, "Advances in evolutionary algorithms," in *Theory, Design and Practice*. Berlin, Germany: Springer, 2006.
[44] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," School Comput. Sci. Electr. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. 20091, 2009.
[45] M. Liu, X. Zou, Y. Chen, and Z. Wu, "Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances," in *Proc. IEEE Congr. Evol. Comput.*, Trondheim, Norway, 2009, pp. 2913–2918.
[46] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
[47] J. Derrac, S. Garca, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.
[48] J. Durillo *et al.*, "A study of multiobjective metaheuristics when solving parameter scalable problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 618–635, Aug. 2010.
[49] L. Marti, J. Garcia, A. Berlanga, and J. Molina, "Introducing MONEDA: Scalable multiobjective optimization with a neural estimation of distribution algorithm," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Atlanta, GA, USA, 2008, pp. 689–696.
[50] L. Marti, J. Garcia, A. Berianga, and J. Molina, "Multi-objective optimization with an adaptive resonance theory-based estimation of distribution algorithm," *Ann. Math. Artif. Intell.*, vol. 68, no. 4, pp. 247–273, 2013.
[51] H. Wang, L. Jiao, R. Shang, S. He, and F. Liu, "A memetic optimization strategy based on dimension reduction in decision space," *Evol. Comput.*, vol. 23, no. 1, pp. 69–100, 2015.
[52] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.
[53] Y. Qi, F. Liu, M. Liu, M. Gong, and L. Jiao, "Multi-objective immune algorithm with Baldwinian learning," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2654–2674, 2012.
[54] J. He, T. Chen, and X. Yao, "On the easiest and hardest fitness functions," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 295–305, Apr. 2015.

[55] G. Karafotias, M. Hoogendoorn, and A. Eiben, "Parameter control in evolutionary algorithms: Trends and challenges," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 167–187, Apr. 2015.

[56] W. Song, Y. Wang, H. Li, and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 414–431, Jun. 2015.

[57] T. Tusar and B. Filipic, "Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 225–245, Apr. 2015.

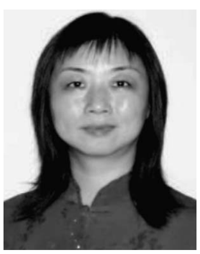
[58] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

[59] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 103–127, 2015.

[60] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 400–413, Jun. 2015.

Xiaoliang Ma was born in Zhejiang, China, in 1984. He received the B.S. degree in computing computer science and technology from Zhejiang Normal University, Jinhua, China, in 2006. He is currently working toward the Ph.D. degree with the School of Computing, Xidian University, Xi'an, China.

His research interests include evolutionary computation, multiobjective optimization, complex network, cooperative coevolution, and bioinformatics.



Fang Liu (SM'07) was born in Beijing, China, in 1963. She received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1984 and the M.S. degree from Xidian University, Xi'an, in 1995, both in computer science and technology.

She is a Professor with Xidian University. Her research interests include signal and image processing, nonlinear circuit and systems theory, learning theory and algorithms, optimization problems, wavelet theory, and data mining.

Yutao Qi was born in Henan, China, in 1981. He received the B.S. degree in software engineering from the School of Software Engineering, Xidian University, Xi'an, China, in 2003 and the M.S. degree in computer science and technology from the Institute of the Information Processing, Xidian University, in 2006.

He is an Associate Professor with Xidian University. His research interests include evolutionary computation, multiagent systems, artificial immune systems, parallel computing, and data mining.

Xiaodong Wang received the B.S. degree in applied mathematics from the Harbin Institute of Technology, Harbin, China, in 1998, and the M.S. degree in computational mathematics from the Inner Mongolia University of Technology, Hohhot, China, in 2007. He is currently working toward the Ph.D. degree in computer application with the School of Computer Science and Technology and the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, China.

His research interests include convex optimization, compressive sensing, and pattern recognition.

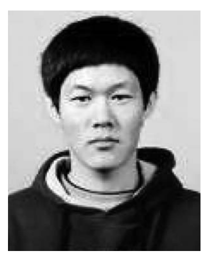
Lingling Li received the B.S. degree in electronic science and technology from the School of Electronic Engineering, Xidian University, Xi'an, China, in 2011, where she is currently working toward the Ph.D. degree in pattern recognition and intelligent system with the School of Electronic Engineering.

Her research interests include community detection in networks and multiobjective optimization.

Licheng Jiao (SM'89) received the B.S. degree in electronic science and technology from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees in electronic circuit and system from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

Since 1992, he has been a Professor with the School of Electronic Engineering, Xidian University, Xi'an, where he is currently the Director of Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education of China. His research interests include image processing, natural computation, machine learning, and intelligent information processing. He is in charge of about 40 important scientific research projects and has published over 20 monographs and 100 papers in international journals and conferences.

Prof. Jiao is the President of Computational Intelligence Chapter of the IEEE Xi'an Section and the IET Xi'an Network, the Chairman of the Awards and Recognition Committee, the Vice Board Chairperson of the Chinese Association of Artificial Intelligence, a Councilor of the Chinese Institute of Electronics, a Committee Member of the Chinese Committee of Neural Networks, an Expert of the Academic Degrees Committee of the State Council, and a member of the IEEE Xi'an Section Execution Committee.



Minglei Yin was born in Shandong, China, in 1990. He received the B.S. degree in computing computer science and technology from Dalian University, Dalian, China, in 2013. He is currently working toward the M.Sc. degree with the School of Computing, Xidian University, Xi'an, China.

His research interest include evolutionary computation.

Maoguo Gong (M'07) received the B.S. degree in electronic engineering (First Class Hons.) and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively.

Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted to an Associate Professor and a Full Professor, respectively, both with exceptive admission. He is currently a Full Professor with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University. His research interest include computational intelligence with applications.

Prof. Gong received the New Century Excellent Talent in the University of the Ministry of Education of China, the Eighth Young Scientist Award of Shaanxi, the New Scientific and Technological Star of Shaanxi Province, and the Science and Technology Award of Shaanxi Province (First Level, 2008 and 2010). He is an Executive Committee Member of the Natural Computation Society of Chinese Association for Artificial Intelligence and a Senior Member of the Chinese Computer Federation.