Tri-Goal Evolution Framework for Constrained Many-Objective Optimization

Yalan Zhou, Min Zhu, Jiahai Wang, *Member, IEEE*, Zizhen Zhang, Yi Xiang, and Jun Zhang, *Fellow, IEEE*

***Abstract*—It is generally accepted that the essential goal of many-objective optimization is the balance between convergence and diversity. For constrained many-objective optimization problems (CMaOPs), the feasibility of solutions should be considered as well. Then the real challenge of constrained many-objective optimization can be generalized to the balance among convergence, diversity, and feasibility. In this paper, a tri-goal evolution framework is proposed for CMaOPs. The proposed framework carefully designs two indicators for convergence and diversity, respectively, and converts the constraints into the third indicator for feasibility. Since the essential goal of constrained many-objective optimization is to balance convergence, diversity, and feasibility, the philosophy of the proposed framework matches the essential goal of constrained many-objective optimization well. Thus, it is natural to use the proposed framework to deal with CMaOPs. Further, the proposed framework is conceptually simple and easy to instantiate for constrained many-objective optimization. A variety of balance schemes and ranking methods can be used to achieve the balance among convergence, diversity and feasibility. Three typical instantiations of the proposed framework are then designed. Experimental results on a constrained many-objective optimization test suite show that the proposed framework is highly competitive with existing state-of-the-art constrained many-objective evolutionary algorithms for CMaOPs.**

***Index Terms*—Constrained many-objective optimization, constraint handling, convergence, diversity, feasibility, tri-goal evolution (TiGE).**

I. Introduction

Constrained many-objective optimization problems (CMaOPs) involve the optimization of more than three conflicting objectives simultaneously subject to constraints [1], [2]. In the real world, CMaOPs are common and occur in many scientific and engineering applications [1], [3]–[7]. Compared with unconstrained many-objective optimization problems (MaOPs), some complicated features are presented in CMaOPs and it is much more difficult for CMaOPs to be solved. Constraint functions greatly increase the difficulty of optimizing problems.

Some major effects of the constraints on Pareto front (PF) in CMaOPs include [8]: 1) infeasibility makes the original unconstrained PF partially feasible; 2) infeasibility comes on the way of converging to the PF so that the search of algorithms will not have a smooth way to approach the PF; 3) the original PF is no more completely feasible and every constrained Pareto optimal solution lies on some constraint boundaries; and 4) constraints may reduce the dimensionality of the PF. Constraints define the infeasible regions in the search space, leading to different types of difficulty for the resulting CMaOPs. Three primary difficulty types of constraints are identified as convergence-hardness, diversity-hardness, and feasibility-hardness [8]. Generally, the PFs of CMaOPs with diversity-hardness constraints have a number of discrete segments, or the diversity-hardness constraints make the search space imbalanced, which means that some parts of the PFs are more difficult to be achieved than other parts. For constraints with feasibility-hardness, the ratios of feasible regions in the search space are usually very low. It is difficult for a constrained many-objective evolutionary algorithm (CMaOEA) to generate a feasible solution on the feasibility-hard CMaOPs, and most solutions in the population are usually infeasible in the initial stage. CMaOPs with convergence-hard constraints hinder the convergence of CMaOEAs to their PFs. Infeasibility comes on the way of converging to the PF so that algorithms can not approach the PF smoothly.

CMaOEAs [3], [9], [10] are considered promising in solving CMaOPs. Essentially, two main issues need to be handled carefully when designing a CMaOEA. One is to balance the feasible solutions with worse objective values and the

Manuscript received May 27, 2018; accepted July 17, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61673403 and Grant U1611262, and in part by the China Scholarship Council under Grant 201608440449. This paper was recommended by Associate Editor Z. Wang. *(Corresponding author: Jiahai Wang.)*

Y. Zhou is with the College of Information, Guangdong University of Finance and Economics, Guangzhou 510320, China.

M. Zhu and Z. Zhang are with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China.

J. Wang is with the Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, China, also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Sun Yat-sen University, Guangzhou 510275, China, and also with the Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510275, China (e-mail: wangjiah@mail.sysu.edu.cn; wjiahai@hotmail.com).

Y. Xiang is with the School of Software, South China University of Technology, Guangzhou 510006, China.

J. Zhang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. This includes a PDF containing tables and figures not presented in the paper. This material is 80.34 MB in size.

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/TSMC.2018.2858843

infeasible solutions with better objective values, the other is to balance the convergence and diversity of a CMaOEA.

For the balance between the feasible solutions with worse objective values and the infeasible solutions with better objective values, constraint-handling techniques (CHTs) are designed by researchers. In fact, the challenge for the balance is the way of using infeasible solutions [2], [11]. Generally, CHTs can be classified into five different groups [12], [13], which is detailed in Section II-B.

For the balance between convergence and diversity, multiobjective evolutionary algorithms (MOEAs) are designed by researchers. MOEAs can be classified into three categories based on the selection strategies, including dominance-based, decomposition-based, and indicator-based methods. Pareto dominance-based methods (e.g., NSGA-II [14], SPEA2 [15], and others [16], [17]) survive solutions into the next generation in terms of the nondominated rank and density estimation. The popular NSGA-II improves the performance of convergence with nondominated sorting, and maintains the performance of diversity with crowding distance. For the group of indicator-based methods, the fitness of a solution depends on its contribution to the performance metrics. Such methods include SMS-EMOA [18] and HypE [19]. For the third type of decomposition-based methods (e.g., MOEA/D [20], MOEA/D-M2M [21], MOEA/D-CRA [22], D-NSGAII [1], MOEA/DD [9], and NSGA-III [23]), an MOP is decomposed into a number of subproblems by a predefined set of uniformly distributed weight vectors, which can be solved collaboratively. Recently, bi-goal evolution (BiGE) [24] is proposed for MaOPs. In BiGE, two observations are discussed: 1) the conflict between the convergence and diversity requirements is aggravated with the increase of the number of objectives and 2) the Pareto dominance loses its effectiveness for a high-dimensional objective space but works well for a low-dimensional objective space. To handle the aforementioned conflict is the second challenge of CMaOPs.

Inspired by the above discussions and BiGE algorithm, this paper presents a tri-goal evolution (TiGE) framework for CMaOPs. TiGE carefully designs two indicators for convergence and diversity, respectively, and converts the constraints into the third indicator for feasibility. A variety of balance schemes and ranking methods can be used to achieve the balance among convergence, diversity, and feasibility. Three instantiations of TiGE, named TiGE-1, TiGE-2, and TiGE-3, are then designed. Experimental results on the constrained many-objective optimization test suite DAC-MaOP1-9 [8] show that TiGE is highly competitive with existing state-of-the-art CMaOEAs for CMaOPs.

The main contributions of this paper are threefold.

1) This paper presents a general framework, TiGE, for CMaOPs. Realizing that the essential challenge for CMaOPs is the balance among convergence, diversity, and feasibility, TiGE constructs a tri-goal model for CMaOPs and adopts different balance schemes to solve the tri-goal problem.
2) Three instantiations of TiGE framework are implemented. In these instantiations, two kinds of balance schemes with different ranking methods are used.
3) This paper provides a comprehensive comparison with state-of-the-art CMaOEAs on the benchmark functions proposed recently.

The rest of this paper is organized as follows. In Section II, the background of this paper is introduced. TiGE is detailed in Section III. Section IV presents three instantiations of TiGE. Section V gives the experiments. Finally, the conclusions are drawn in Section VI.

II. Background

A. Basic Concepts

CMaOPs optimize more than three conflicting objectives with many constraints to be fulfilled. Generally, a CMaOP can be stated as follows:

$$\begin{aligned} \text{minimize} \quad & F(X) = (f_1(X), \ldots, f_M(X))^T \\ \text{subject to} \quad & g_j(X) \geq 0, j = 1, \ldots, J \\ & h_k(X) = 0, k = 1, \ldots, K \\ & X \in \Omega \end{aligned} \tag{1}$$

where $F(X)$ is an M-dimensional objective vector, $g_j(X) \geq 0$ and $h_k(X) = 0$ define J inequality constraints and K equality constraints, respectively. $F : \Omega \rightarrow \Theta \subseteq R^M$ maps decision space Ω to M-dimensional attainable objective space Θ.

Feasible: A solution X is said to be *feasible* if all constraints are satisfied; otherwise it is *infeasible*.

Feasible Set (FS): The FS consists of all feasible solutions in decision space.

Pareto Dominance: For two feasible solutions X^1 and X^2, X^1 is said to dominate X^2, denoted by $X^1 \preceq X^2$, iff $\forall i \in \{1, 2, \ldots, M\}, f_i(X^1) \leq f_i(X^2)$ and $\exists j \in \{1, 2, \ldots, M\}, f_j(X^1) < f_j(X^2)$. If two solutions neither dominate each other, then it is said that they are *nondominated*.

Pareto Optimal: A feasible solution $X^* \in \text{FS}$ is said to be *Pareto optimal* iff $\nexists X \in \text{FS}, X \preceq X^*$.

Pareto Set (PS): The set of all Pareto optimal solutions is called the PS.

PF: The PF is defined as: $\text{PF} = \{F(X) \in \Theta | X \in \text{PS}\}$.

B. Constraint-Handling Approaches

Many CHTs have been designed to balance the objective function and constraints for constrained single-objective optimization problems (CSOPs). To achieve the optimization of CMaOPs, the CHTs in this paper need to be appropriately adapted. The existing CHTs can be classified into five different groups, including feasibility maintenance, penalty functions, separation of constraints and objectives, multiobjective constraint handling, and hybrid methods [12], [13]. Recently, some other CHTs are proposed in [25]–[28]. Several methods closely related to this paper are detailed below.

1) Penalty Functions: By adding or subtracting a certain value to or from the objective function based on the amount of constraint violation, the penalty function method converts a constrained optimization problem into an unconstrained one. Generally, two kinds of penalty methods are considered: 1) exterior and 2) interior [29]. Interior method, also called barrier method, searches solutions within feasible region. The penalty function value will be small at points away from the constraint boundaries and will tend to infinity as the constraint boundaries are approached. There are different types of penalty functions used with EAs, such as death penalty, static penalty, dynamic penalty, adaptive penalty, and so on. Besides, some other constraint-handling approaches that use penalty functions, like self-adaptive fitness formulation [30], ASCHEA [31] deserve special consideration.

2) Stochastic Ranking: Stochastic ranking [32] can be taken as a special variant of penalty function method. The basic idea of the approach is to utilize a predefined probability parameter p_f to balance the influence of the objective function and the penalty function when assigning fitness to an individual. Stochastic ranking is implemented in the population of an EA, where the comparison between two adjacent individuals based on either the objective function or the sum of constraint violation is stochastic and dependent on p_f. It is just as the bubble sort procedure of an array.

3) Multiobjective Constraint Handling: In this approach, the $J + K$ constraints can be converted into another $J + K$ objective functions as follows [33]:

$$\begin{aligned} G_j(X) &= \max\{0, g_j(X)\},\ j = 1, \ldots, J \\ H_k(X) &= \max\{0, |h_k(X)| - \delta\},\ k = 1, \ldots, K \end{aligned} \tag{2}$$

or simply one objective function as follows [23]:

$$f_{\mathrm{cv}}(X) = \sum_{j=1}^{J} \langle \overline{g}_j(X) \rangle + \sum_{k=1}^{K} \left| \overline{h}_k(X) \right| \tag{3}$$

where δ is a small positive tolerance value, $\overline{g}_j(X)$ and $\overline{h}_k(X)$ are normalized constraint functions. The bracket operator $\langle \alpha \rangle$ returns the absolute value of α if $\alpha < 0$, and returns 0 otherwise. Thus, the original CSOP is transformed into the following unconstrained bi-objective optimization with (3):

$$\text{minimize} \quad (f(X), f_{\mathrm{cv}}(X)). \tag{4}$$

4) Unbiased Bi-Objective Optimization Model: Although the idea of multiobjective constraint handling is very attractive, a search bias toward the feasible region must still be introduced in optimization if a feasible solution is to be found. When comparing multiobjective constraint handling with penalty function method, it becomes evident that multiobjective constraint handling does not work as well as one might first think [34]. It does not solve the fundamental problem of balancing the objective function and constraint violation faced by the penalty function approach. Therefore, extra mechanisms should be integrated into multiobjective constraint handling to make a good balance between both objectives in (4). A search bias is introduced into an MOEA by defining the b-dominance relation in [35]. However, adding other mechanisms also results in the more complexity of the approach simultaneously.

To avoid introducing the additional mechanisms into an MOEA, an unbiased bi-objective optimization model is proposed in [36], which converts the bi-objective optimization of (4) into a new one as follows:

$$\text{minimize} \quad (f(X) + \epsilon f_{\mathrm{cv}}(X), f_{\mathrm{cv}}(X)) \tag{5}$$

where $\epsilon f_{\mathrm{cv}}(X)$ can be regarded as a penalty term to penalize the objective function and $\epsilon > 0$ is the penalty factor. ϵ is used to adjust the preference to constraints. On the one hand, ϵ should tend to infinity to ensure that the Pareto optimal solution of model (5) is exactly the optimum of the original CSOP. On the other hand, too large ϵ may overemphasize the preference to constraint violation, which weakens the effect of the objective function. Thus, a gradually increased ϵ in the evolutionary process is well-suited for the balance between the objective function and constraints. An easy way to update ϵ is given by

$$\epsilon(t+1) = \rho\epsilon(t) \tag{6}$$

where t is the generation in the evolution and $\rho > 1$ is the scaling factor.

As a result, the new optimization model of (5) can be solved with a generic MOEA without constraint handling, thus many popular unconstrained MOEAs can be applied to it. The novel unbiased model will be adopted in this paper due to its simplicity and universality.

C. CMaOEAs and Benchmark Problems

Much more attention has been restricted with unconstrained MaOPs [23], [37]–[39]. There is few literature on dealing with CMaOPs. Representative CMaOEAs for CMaOPs, including CNSGA-III [3], CMOEA/DD [9], CVaEA [10] are reviewed as follows. Compared with the original NSGA-III [23], a constrained dominance principle is applied to nondominated sorting in CNSGA-III to deal with the constraints. In addition, a modified tournament selection is used in CNSGA-III for the mating selection, which prefers the individual with smaller constraint violation. In CMOEA/DD, the feasible solutions will survive to the next round without reservation while the survival of infeasible ones depends on both constraint violation and niching scenarios. Infeasible solution associated with an isolated subregion is given a second chance to survive. As for CVaEA, a modified tournament selection operation as in CNSGA-III is used to choose mating parents and a portion of infeasible solutions is added into the new population before the inclusion of feasible ones. In the presence of infeasible solutions, the modified worse-elimination principle emphasizes feasible solutions over infeasible ones, and emphasizes solutions with smaller fitness values (i.e., the sum of all normalized objectives) over those with larger values.

Further, there are still few explicit and precise definitions of constraint difficulty in CMaOPs, which make it difficult to evaluate the performance of CMaOEAs tested on them [40]. In [8], three primary types of difficulty to characterize the constraints in CMaOPs, including feasibility-hardness, convergence-hardness, and diversity-hardness, are first proposed. Then a general toolkit is developed to construct difficulty adjustable CMaOPs (DAC-MaOPs) with three types of parameterized constraint functions according to the proposed three primary types of difficulty. To demonstrate the scalability to the number of objectives in CMaOPs, nine DAC-MaOPs, named DAC-MaOP1-9, with the scalability to the number of objectives are also suggested. To

Algorithm 1: BiGE

```
Input: population 𝒫, population size N
Output: population 𝒫
begin
    𝒫 = initialization(𝒫);
    while stopping criterion is not fulfilled do
        proximityEstimation(𝒫);
        crowdingDegreeEstimation(𝒫);
        𝒫′ = matingSelection(𝒫);
        𝒫″ = variation(𝒫′);
        𝒬 = 𝒫 ∪ 𝒫″;
        𝒫 = environmentalSelection(𝒬);
    end while
end
```

verify the effectiveness of TiGE framework, this paper adopts DAC-MaOPs as the benchmark problems.

D. Brief Review of BiGE

The idea of BiGE [24] is motivated by the two observations mentioned in Section I. BiGE maps solutions from the original objective space to a bi-goal objective space (proximity and crowding degree), within which Pareto nondominated sorting method [14] is adopted as the balance scheme. The pseudo code of BiGE is shown in Algorithm 1.

First, the proximity of an individual is measured by the sum of its normalized objectives

$$f_{\mathrm{pr}}(X) = \sum_{k=1}^{M} f_k(X) \tag{7}$$

where $f_k(X)$ is the normalized objective value of individual X in the kth objective. Then $f_{\mathrm{pr}}(X)$ could reflect the convergence performance of each individual. A smaller f_{pr} value of an individual usually indicates better performance of convergence.

Then, the crowding degree of an individual is defined as

$$f_{\mathrm{cd}}(X) = \sqrt{\sum_{Y \in P, Y \neq X} \mathrm{sh}(X, Y)} \tag{8}$$

where $\mathrm{sh}(X, Y)$ denotes a sharing function between two individuals X and Y based on niching technique. $\mathrm{sh}(X, Y)$ is defined as

$$\mathrm{sh}(X, Y) = \begin{cases} \left(0.5\left(1 - \dfrac{d(X, Y)}{r}\right)\right)^2, & \text{if } d(X, Y) < r, f_{\mathrm{pr}}(X) < f_{\mathrm{pr}}(Y) \\ \left(1.5\left(1 - \dfrac{d(X, Y)}{r}\right)\right)^2, & \text{if } d(X, Y) < r, f_{\mathrm{pr}}(X) > f_{\mathrm{pr}}(Y) \\ \mathrm{rand}(), & \text{if } d(X, Y) < r, f_{\mathrm{pr}}(X) = f_{\mathrm{pr}}(Y) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $d(X, Y)$ is the Euclidean distance between objective vectors of two individuals X and Y. The function rand() means to assign either $\mathrm{sh}(X, Y) = (0.5(1 - [d(X, Y)/r]))^2$ or $\mathrm{sh}(X, Y) = (1.5(1-[d(X, Y)/r]))^2$ randomly. r denotes a niche radius simply depends on the population size N as well as the number of objectives M, that is, $r = 1/\sqrt[M]{N}$.

There is a strong negative correlation between the crowding degree of an individual and its Euclidean distance to other individuals in the population. That is, a larger Euclidean distance to other individuals brings a certain individual lower crowding degree, which implies better diversity performance. The binary tournament selection based on Pareto dominance in bi-goal domain is adopted in the mating selection. Nondominated sorting method is adopted in environmental selection to deal with the optimization problem in bi-goal domain, as in NSGA-II for bi-objective problem. Note that in the critical layer, individuals are randomly selected for simplicity.

As in BiGE, two indicators for convergence and diversity, respectively, are defined in stochastic ranking-based multi-indicator algorithm (SRA) [41] for MaOPs. SRA adopts the stochastic ranking technique to balance the search biases of different indicators and thus convergence and diversity.

III. Proposed Framework: TiGE

A. Motivation

It is generally accepted that the final goal of MaOPs is to obtain good balance between convergence and diversity as shown in BiGE. Thus, we can further consider that the essential goal of CMaOPs is to balance the triplet: convergence, diversity, and feasibility. Thus, this paper tries to extend the idea of BiGE to deal with CMaOPs.

To achieve optimization of convergence and diversity, the M-dimensional objective space of an MaOP can be mapped to a bi-goal objective space related to them. When constraints are introduced, feasibility should be considered as well, where the third goal for feasibility is obtained with some constraint handling techniques. So far, a tri-goal optimization problem related to the CMaOP is presented. With some balance schemes, a new CMaOEA, called TiGE, is proposed as follows.

B. General Framework

The main procedure of TiGE is described in Algorithm 2. First of all, a population $\mathcal{P}$ with N individuals is initialized randomly. Second, the three indicators for convergence, diversity and feasibility are estimated, respectively. Then, a set of promising individuals are selected from $\mathcal{P}$ and put into the mating pool, after which variation operators are implemented and an offspring population is generated. Finally, environmental selection survives N individuals into the next generation in terms of the three indicators. As stated in Algorithm 3, TiGE utilizes a simple binary tournament selection strategy to choose good individuals for mating selection. It should be noted that f_{fitness} is calculated in environmental selection, and the value of f_{fitness} is initialized to 0. The variation operations are not fixed in TiGE and can be freely chosen by users.

C. Environmental Selection

Algorithm 4 describes the environmental selection procedure of TiGE. First, a normalization function, normalization($\mathcal{Q}$), is implemented for each objective of individuals with its minimum and maximum values in the union population $\mathcal{Q}$. Following the idea of utilizing two indicators to balance convergence and diversity in BiGE, TiGE adopts three indicators to balance convergence, diversity, and

Algorithm 2: TiGE

Input: population $\mathcal{P}$, population size N
Output: population $\mathcal{P}$

begin
  $\mathcal{P}$ = initialization($\mathcal{P}$);
  while stopping criterion is not fulfilled **do**
    convergenceEstimation($\mathcal{P}$);
    diversityEstimation($\mathcal{P}$);
    feasibilityEstimation($\mathcal{P}$);
    $\mathcal{P}'$ = matingSelection($\mathcal{P}$);
    $\mathcal{P}''$ = variation($\mathcal{P}'$);
    $\mathcal{Q} = \mathcal{P} \cup \mathcal{P}''$;
    $\mathcal{P}$ = environmentalSelection($\mathcal{Q}$);
  end while
end

Algorithm 3: TiGE: Tournament Selection

Input: population $\mathcal{P}$
Output: individual X

begin
  if $f_{fitness}(X^1) < f_{fitness}(X^2)$ **then**
    $X = X^1$;
  else if $f_{fitness}(X^1) > f_{fitness}(X^2)$ **then**
    $X = X^2$;
  else if random(0,1) < 0.5 **then**
    $X = X^1$;
  else
    $X = X^2$;
  end if
end

Algorithm 4: TiGE: environmentalSelection($\mathcal{Q}$)

Input: union population $\mathcal{Q}$, population size N
Output: population $\mathcal{P}$

begin
  Generate an empty population $\mathcal{P}$;
  normalization($\mathcal{Q}$);
  foreach $X \in \mathcal{Q}$ **do**
    $f_c(X)$ = convergenceEstimation(X);
    $f_d(X)$ = diversityEstimation(X);
    $f_f(X)$ = feasibilityEstimation(X);
  end foreach
  $f_{fitness}(X) = \underset{X \in \mathcal{Q}}{\text{balanceScheme}}(f_c(X), f_d(X), f_f(X))$ with ranking method;
  select the best N individuals by the fitness function $f_{fitness}$ and add them into population $\mathcal{P}$;
end

feasibility. In this paper, the constraint violation f_{cv} is used to measure feasibility of an individual, i.e., $f_f = f_{\text{cv}}$. Then the three indicators of individuals are estimated. Next, a balance scheme is applied to the new tri-goal problem whose fitness is denoted as f_{fitness}. Finally, the best N individuals are selected from the union population by fitness function f_{fitness} and added into population $\mathcal{P}$.

D. Balance Schemes

As mentioned before, a search bias toward constraints should be introduced for CMaOPs. The key challenge is to deal with the coordination between constraints and objectives. In what follows, two kinds of balance schemes are designed for TiGE.

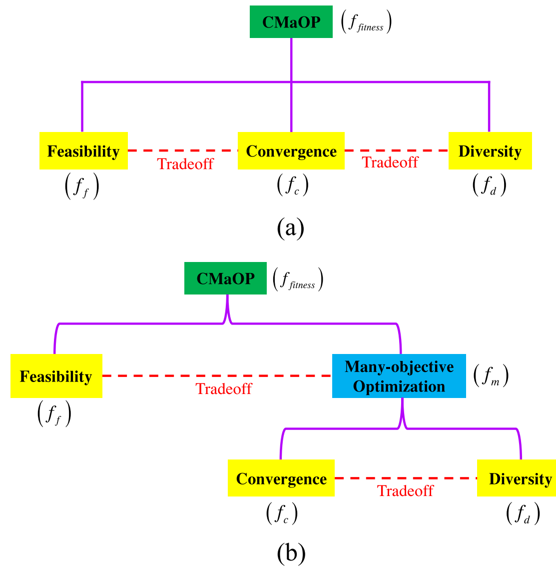


Fig. 1. Design principle of balance schemes. (a) One-level balance scheme. Three goals are to be handled simultaneously with multiobjective optimizers. (b) Two-level balance scheme. The balance between convergence and diversity is managed first, and then the balance between many-objective optimization and feasibility is processed.

BiGE treats the two goals equally and optimizes them simultaneously for MaOPs, which suggests that it could be extended to handle the three goals simultaneously with multiobjective optimizers for CMaOPs. The new strategy is called the one-level balance scheme. Its design principle is depicted in Fig. 1(a).

Convergence and diversity are both derived from objectives, and feasibility is derived from constraints. Convergence and diversity make sense only when feasibility is satisfied. Thus, the ultimate goal of CMaOPs should be the balance between convergence and diversity under the premise of feasibility. Hence, a two-level balance scheme is proposed. Its design principle is depicted in Fig. 1(b). The bottom level focuses on the balance between convergence and diversity, then the upper focuses on the balance between many-objective optimization and feasibility.

E. Characteristics of TiGE

The characteristics of TiGE are summarized as follows.

1) TiGE is a new general framework that allows instantiating different instantiations by using different combination of components from the same framework. Thus, it supports automatic component-wise design of new CMaOEAs [42]. Table I shows the algorithmic component options available for TiGE framework. By selecting different values of each algorithmic component, different algorithms can be instantiated. Table II shows how to instantiate three algorithms considered in this paper. In Section IV, three instantiations of TiGE are given in detail.
2) TiGE is a simple and general framework. A large amount of existing multiobjective algorithms for bi-

TABLE I
ALGORITHMIC COMPONENT OPTIONS AVAILABLE FOR TIGE FRAMEWORK

Component	Domain	Reference
Indicators for tri-goal (f_c, f_d, f_f)	(f_{pr}, f_{cd}, f_{cv});(I_1, I_2, f_{cv})	[24], [41]
Balance (Tradeoff) scheme	One-level: ranking(f_c, f_d, f_f); Two-level: ranking(ranking(f_c, f_d), f_f)	Section IV
Ranking (Sorting) method	Nondominated sorting (NDS); Stochastic ranking (SR)	[14], [41]

TABLE II
INSTANTIATION OF TIGE FRAMEWORK USING DIFFERENT COMBINATION OF COMPONENTS

Algorithm	Indicator	Balance scheme	Ranking method	Reference
TiGE-1	(f_{pr}, f_{cd}, f_{cv})	One-level: ranking(f_c, f_d, f_f)	Nondominated sorting (NDS)	Section IV-A
TiGE-2	(f_{pr}, f_{cd}, f_{cv})	Two-level: ranking(ranking(f_c, f_d), f_f)	Nondominated sorting (NDS)	Section IV-B
TiGE-3	(I_1, I_2, f_{cv})	Two-level: ranking(ranking(f_c, f_d), f_f)	Stochastic ranking (SR)	Section IV-C

and tri-objective problems can be adapted to tackle the tri-goal problem proposed in this paper.

3) The essential goal of CMaOPs is to balance the triplet: convergence, diversity, and feasibility. The philosophy of TiGE matches the essential goal of CMaOPs well. Thus, it is natural to use TiGE to deal with CMaOPs. Further, TiGE is conceptually simple and easy to instantiate for CMaOPs.

IV. INSTANTIATIONS OF TIGE

By carefully selecting the values of each algorithmic component, three representative instantiations, TiGE-1, TiGE-2, and TiGE-3, can be designed. Both BiGE and SRA have a simple structure and are demonstrated to be effective for MaOPs. Thus, components from both BiGE and SRA are selected for the instantiations. Table II shows how to instantiate three algorithms considered in this paper.

A. TiGE-1

1) Indicators for Tri-Goal: First, the two indicators for many-objective optimization can be formulated as in BiGE. Then, the indicator for feasibility is measured by constraint violation f_{cv} in TiGE, which is calculated by (3). The indicators are defined as follows:

$$f_c(X) = f_{\text{pr}}(X) \tag{10}$$
$$f_d(X) = f_{\text{cd}}(X) \tag{11}$$
$$f_f(X) = f_{\text{cv}}(X). \tag{12}$$

2) Ranking Method: TiGE-1 is designed as an instantiation with the one-level balance scheme. Inspired by the unbiased bi-objective model [36], the balance model of TiGE-1 is presented as follows:

$$\begin{aligned} &\text{minimize} \quad (f_c(X) + \epsilon f_{\text{cv}}(X), f_d(X), f_{\text{cv}}(X))^T \\ &\text{subject to} \quad X \in \Omega. \end{aligned} \tag{13}$$

Some generic MOEAs can be used to solve the new optimization model. Since nondominated sorting is a classic ranking strategy for MOPs with 2 or 3 objectives, it is natural to adopt nondominated sorting as the ranking method for the balance model defined by (13). Here, the individuals with smaller rank values (denoted as f_{rank}) are preferred, otherwise individuals with smaller f_{cv} are preferred when they are in the same layer or have the same rank value. The fitness function $f_{\text{fitness}}(X)$ obtained by the ranking rule mentioned above can be mathematically formulated as follows:

$$f_{\text{fitness}}(X) = f_{\text{rank}}(X) + \frac{f_{\text{cv}}(X)}{f_{\text{cv}}(X) + 1} \tag{14}$$

where $f_{\text{rank}}(X)$ indicates the rank value of individual X obtained by nondominated sorting.

Equation (14) is explained as follows. f_{rank} is a positive integer serving as a dominant part. $[f_{\text{cv}}/(f_{\text{cv}}+1)]$ is in the range [0, 1) due to $f_{\text{cv}} \geq 0$ serving as a subdominant part. For two individuals X and Y, if $f_{\text{rank}}(X) < f_{\text{rank}}(Y)$, then $f_{\text{fitness}}(X) < f_{\text{fitness}}(Y)$ regardless of the relation between the subdominant parts. When $f_{\text{rank}}(X) = f_{\text{rank}}(Y)$, a smaller f_{cv} will result in better f_{fitness} since $[f_{\text{cv}}/(f_{\text{cv}} + 1)]$ is a monotonic increasing function of f_{cv}.

3) Complexity Analysis: Attention should be paid to the main steps in environmental selection when analyzing the computational complexity of TiGE-1. First, the time complexity for normalization is $O(MN)$. Due to the computation of Euclidean distance among all individuals and multiobjective constraint handling of (3), the time complexity for estimation of the indicators is $O(MN^2 + (J+K)N)$. Though the effective approach in [43] is adopted, nondominated sorting still has a worst complexity of $O(MN^2)$ and it is definitely $O(N^2)$ for new the balance model of TiGE-1. To summarize, the overall computational complexity of TiGE-1 within one generation is $O(MN^2 + (J + K)N)$.

B. TiGE-2

1) Indicators for Tri-Goal: TiGE-2 also utilizes the three indicators f_{pr}, f_{cd}, and f_{cv}, as defined in (10)–(12), to measure convergence, diversity, and feasibility, respectively.

2) Ranking Method: As a straightforward extension of BiGE for CMaOPs, TiGE-1 tries to balance convergence, diversity, and feasibility with a one-level balance scheme. Further, TiGE-2 is designed as an instantiation with a two-level balance scheme based on TiGE-1, which is described in Fig. 2(a). In TiGE-2, nondominated sorting strategy is adopted as the ranking method of the first-level balance between convergence and diversity, which is presented as follows:

$$\begin{aligned} &\text{minimize} \quad (f_c(X), f_d(X))^T \\ &\text{subject to} \quad X \in \Omega. \end{aligned} \tag{15}$$

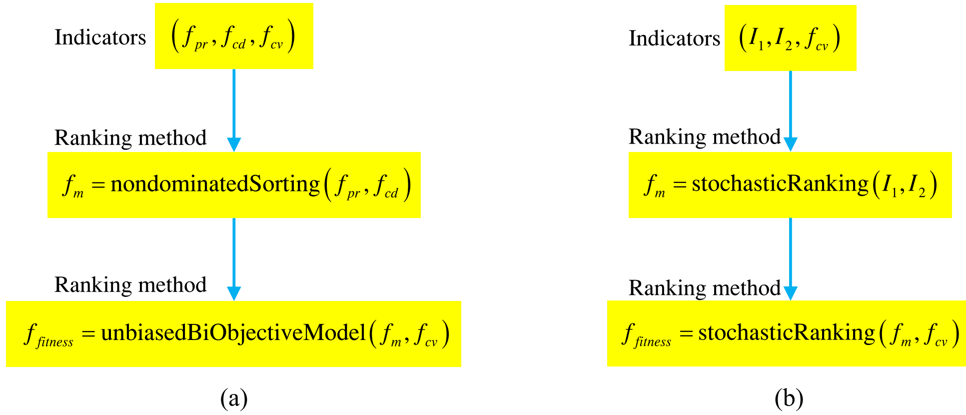


Fig. 2. Two-level balance scheme of TiGE-2 and TiGE-3. (a) TiGE-2. Nondominated sorting is adopted as the ranking method of the first-level balance between convergence and diversity. The unbiased bi-objective model is then used for the second-level balance between many-objective optimization and feasibility. (b) TiGE-3. Stochastic ranking is adopted as the ranking method twice in the two-level balance scheme.

Let the fitness f_m denote the rank value of individuals in the model above. The ranking method of second-level balance is designed to balance many-objective optimization and feasibility. Using the unbiased bi-objective model, the second-level balance model of TiGE-2 is presented as follows:

$$\begin{aligned} &\text{minimize} \quad (f_m(X) + \epsilon f_{\text{cv}}(X), f_{\text{cv}}(X))^T \\ &\text{subject to} \quad X \in \Omega. \end{aligned} \tag{16}$$

Since nondominated sorting is applied to the second-level balance model, individuals with smaller rank values (denoted as f_{rank}) are preferred, otherwise the individuals with smaller f_{cv} are preferred when they are in the same layer or have the same rank value. The fitness function $f_{\text{fitness}}(X)$ obtained by the ranking rule mentioned above can be mathematically formulated as (14).

3) Complexity Analysis: First, the time complexity for normalization is $O(MN)$. Due to the computation of Euclidean distance among all individuals and multiobjective constraint handling of (3) in TiGE-2, the time complexity for estimation of the indicators is $O(MN^2 + (J+K)N)$. Nondominated sorting strategy is adopted twice in the two-level balance scheme. Though the effective approach in [43] is adopted, nondominated sorting still has a worst complexity of $O(MN^2)$ and it is definitely $O(N^2)$ for the optimization of (15) and (16). To summarize, the overall computational complexity of TiGE-2 within one generation is $O(MN^2 + (J+K)N)$.

C. TiGE-3

1) Indicators for Tri-Goal: Different indicators can be used in TiGE. Recently, a many-objective optimization algorithm, SRA [41], introduces other two indicators. The first indicator for convergence is defined as follows:

$$I_{\epsilon+}(X, Y) = \min_{\epsilon}(f_i(X) - \epsilon \le f_i(Y), i \in \{1, \ldots, M\}) \tag{17}$$

$$I_1(X) = \sum_{Y \in \mathcal{Q}, Y \ne X} -e^{-I_{\epsilon+}(X,Y)/0.05}. \tag{18}$$

The second indicator for diversity is defined as

$$I_{\text{SDE}}(X, Y) = \sqrt{\sum_{i=1}^{M} sd(f_i(X), f_i(Y))^2} \tag{19}$$

$$I_2(X) = \min_{Y \in \mathcal{Q}, Y \text{ precedes } X} \{I_{\text{SDE}}(X, Y)\} \tag{20}$$

where

$$sd(f_i(X), f_i(Y)) = \begin{cases} f_i(Y) - f_i(X), & \text{if } f_i(X) < f_i(Y) \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Here, the term "Y precedes X" means that the original index of the position of Y in the population $\mathcal{Q}$ is smaller than X.

The indicator for feasibility in TiGE-3 is detailed as (12). The indicators (I_1, I_2) for convergence and diversity is adopted in TiGE-3

$$f_c(X) = I_1(X) \tag{22}$$

$$f_d(X) = I_2(X). \tag{23}$$

2) Ranking Method: Besides nondominated sorting and unbiased bi-objective model, some other multiobjective optimizers and CHTs can be adopted as ranking methods of the two-level balance scheme. The two-level balance scheme described in Fig. 2(b) is used in TiGE-3, where stochastic ranking is used. First, stochastic ranking is adopted as the ranking method of first-level balance between convergence and diversity. In other words, stochastic ranking algorithm [41] is applied to the two indicators (I_1, I_2). Here, the union population $\mathcal{Q}$ is regarded as an array and the procedure of SR is essentially a bubble sort for $\mathcal{Q}$ based on probability. Specifically, when the bubble sort is executed, the comparison between two adjacent individuals uses indicator I_1 with a probability of p_c or indicator I_2 with a probability of $1 - p_c$. The location of an individual in the array indicates its performance in terms of convergence and diversity. Therefore, the resulting fitness f_m for the indicators (I_1, I_2) is defined as the location index (range from 1 to $|\mathcal{Q}|$) of an individual in the union population.

The method of stochastic ranking [32] is also applied to the second-level balance as follows:

$$\begin{aligned} &\text{minimize} \quad (f_m(X), f_{\text{cv}}(X))^T \\ &\text{subject to} \quad X \in \Omega. \end{aligned} \tag{24}$$

The fitness f_{fitness} of individuals obtained after the second-level balance is defined as the location index (range from 1 to $|Q|$) of an individual in the union population.

3) Complexity Analysis: First, the time complexity for normalization is $O(MN)$. Similar to the former two algorithms, the time complexity for estimation of the indicators is $O(MN^2 + (J+K)N)$ due to the computation of transformed distance among all individuals and multiobjective constraint handling of (3) in TiGE-3. Stochastic ranking strategy is adopted twice in the two-level balance scheme, which has the same complexity of $O(N^2)$ as bubble sort algorithm. To summarize, the overall computational complexity of TiGE-3 within one generation is $O(MN^2 + (J+K)N)$.

V. Experimental Study

Empirical experiments are conducted on a recently proposed test suite, i.e., the DAC-MaOPs [8], to verify the performance of the proposed framework. First, three instantiations of TiGE are tested and compared. Then, three state-of-the-art CMaOEAs for CMaOPs, CMOEA/DD [9], CNSGA-III [3], and CVaEA [10], are used as the competitors for comparison.

A. Test Problems

DAC-MaOP1-9 [8] are recently proposed DAC-MaOPs with the scalability to the number of objectives. As mentioned above, the balance among convergence, diversity and feasibility is the main issue of CMaOEAs. Three primary difficulty types of constraints are identified as convergence-hardness, diversity-hardness, and feasibility-hardness in DAC-MaOPs. Then three parameters (η, ζ, γ) are defined in the range [0, 1] to measure the difficulty level of the three difficulty types, respectively. CMaOPs with different difficulty levels can be constructed with combination of different values of the triplet, which is able to evaluate different abilities of algorithms.

As recommended in [8], the number of objectives $M = 5, 7, 10, 15$ is tested in our experiments. In terms of constraints, the first K constraints are defined to limit the ranges of parameters in the shape functions. The $(K+1)$th constraint is to limit the distance function, which has the ability to control the portion of feasible regions. The last $P+1$ constraints are set on the components of the objective functions, which generate a number of infeasible obstacles near its PF. Two parameters K and P define the total number of constraints. In this paper, $K = M-1$ and $P = M$ are used so that the number of constraints is $2M+1$. In addition, four groups of assignments (0.5, 0, 0), (0, 0.5, 0), (0, 0, 0.5), and (0.5, 0.5, 0.5) to the triplet (η, ζ, γ) are used to test the algorithms.

B. Performance Metrics and Statistics

The inverted generational distance (IGD) [44] and hypervolume (HV) [45] have been frequently used for the comparison of MOEAs. It is argued in [46] that it is not easy to evaluate the difference between the obtained solution set and the PF using distance-based performance measures, such as IGD. This is because 1) Pareto optimal solutions of many-objective problems are usually unknown and 2) a large number of Pareto optimal solutions are needed to calculate the measure in a reliable manner. Thus, only HV is used for performance evaluation in the experiments.

Assume that $r = (r_1, r_2, \ldots, r_M)$ is a reference point that is dominated by all individuals in the approximation set, a hypercube is generated between every individual and the reference point. Then HV value is defined as the volume of the union of the hypercubes. Obviously, both the distribution of the approximation set and its adjacency to the true PF are considered, and HV is a compositive performance indicator. A larger HV value means better comprehensive performance in terms of convergence and diversity. As in [8], $r = (3, 5, \ldots, 2M+1)$ is used as the reference point for DAC-MaOPs in this paper. Due to the high computational complexity of HV, the Monte Carlo method [19] with 1 000 000 sampling points is adopted to approximate the HV value.

Single-problem Wilcoxon rank sum test at 5% significance level is carried out for indicating the significant differences between the two algorithms on a single problem. The statistical results are summarized as $w/t/l$, which imply that the algorithm performs better than, similar to and worse than the competitor on w, t, and l problems, respectively. Additionally, multiproblem Wilcoxon signed rank test at 5% significance level is conducted to distinguish the differences between a pair of algorithms on all problems. A final ranking of all algorithms is given according to Friedman test [47], [48].

Due to limited space, numerical values of HV are presented in Tables S1–S6, in the supplementary material, and feasibility rate of each test problem is presented in Tables S7–S9, in the supplementary material. Note that HV is equal 0 if no feasible solution is found in a certain run. For a total of L runs on each test problem, R_i $(i = 1, 2, \ldots, L)$ is defined as 1 if at least one feasible solution is found in the ith run, or 0 otherwise. Then feasibility rate is calculated as follows:

$$FR = \frac{\sum_{i=1}^{L} R_i}{L}. \tag{25}$$

Statistical results, including $w/t/l$ and ranking values, are shown in Tables III–VII in this paper.

C. Parameter Settings

1) Reproduction Operators: In this paper, rand/1 differential evolution (DE) operator [49], [50] and polynomial mutation (PM) operator [51] are used for all algorithms to generate offspring solutions. $CR = 1.0$ and $F = 0.5$ are set for DE operator; For PM, the distribution index and the mutation probability are set to $\eta_m = 20$ and $p_m = 1/n$ (n is the number of decision variables), respectively.

2) Population Size: For DAC-MaOPs with 5, 7, 10, 15 objectives, the population size of algorithms is set to 210, 210, 220, 135, respectively.

TABLE III
STATISTICS RESULTS OF TIGE-2 WITH TIGE-1 AND TIGE-3 IN TERMS OF HV

TiGE-2 vs	$w/t/l$	$R+$	$R-$	p-value	$\alpha = 0.05$
TiGE-1	108/22/14	9531.0	909.0	<0.05	YES
TiGE-3	126/12/6	10235.0	205.0	<0.05	YES

3) Termination Condition: Each problem is tested 30 times for each algorithm and the number of maximum function evaluations is used as the termination condition of each run. For DAC-MaOPs with 5, 7, 10, and 15 objectives, the number of maximum function evaluations are set to 3×10^5, 4×10^5, 5×10^5, and 6×10^5, respectively.

4) Specific Parameter Settings for Each Algorithm: Two parameters in TiGE-1 and TiGE-2 are required to be predefined, i.e., the penalty factor ϵ used to control the preference of constraints, and ρ to update ϵ at each generation. In the experiment, the initial value of ϵ is set to 0.05 and ρ is set to 1.01 as in [36]. For TiGE-3, two parameters p_c and p_f are to be set for the balance between indicators. p_c is set to a random number generated in [0.4, 0.6] at each generation and p_f is set to 0.45 according to [32] and [41]. Additionally, the penalty-based boundary intersection approach is used in CMOEA/DD with a penalty parameter $\theta = 5$ [8]. The neighborhood size T is set to 20 and the probability used to select in the neighborhood is given by $\delta = 0.9$. In CVaEA, a parameter α indicating the infeasible portion is utilized to control the ratio of infeasible solutions that are preferentially added into the population in the environmental selection. α is set to 0.01 as recommended in [10].

D. Comparison Among TiGE Instantiations

1) TiGE-1 Versus TiGE-2: TiGE-1 is a direct extension of BiGE for CMaOPs, while the balance strategy of TiGE-2 is implemented more subtly with the two-level balance scheme. From Table III, TiGE-2 shows a significant advantage over TiGE-1. TiGE-2 obtains better HV values on 108 out of the 144 test instances compared with TiGE-1. For the multiproblem Wilcoxon signed-rank test, TiGE-2 obtains higher $R+$ value than $R-$ value. Additionally, for TiGE-2 vs TiGE-1, the p value is far less than 0.05, which shows that TiGE-2 is significantly better than TiGE-1. Better performance of TiGE-2 may be due to the two-level balance scheme.

Fig. 3 plots the approximation sets obtained by the TiGE variants on the 10-objective DAC-MaOP4 with different difficulty triplets by parallel coordinates. For the problem with difficulty triplet (0.5, 0.5, 0.5), as shown in Fig. 3(a) and (b), the approximation set obtained by TiGE-2 covers more objectives than TiGE-1, which indicates that TiGE-2 has better performance in terms of diversity. As shown in Fig. 3(d), (e), (g), (h), (j), and (k), TiGE-1 and TiGE-2 show similar performance for remaining cases.

2) TiGE-2 Versus TiGE-3: Since the two-level balance scheme leads to better performance in TiGE-2, it is also adopted in TiGE-3. TiGE-3 is the extension of SRA for CMaOPs. As described in Table III, TiGE-2 generally outperforms TiGE-3. First, TiGE-2 obtains better HV values on 126 out of the 144 test instances compared with TiGE-3. Higher $R+$ value than $R-$ value indicates the advantage of TiGE-2 over TiGE-3. Besides, the significantly small p value in Table III shows that TiGE-2 significantly performs better than TiGE-3. Though the two-level balance scheme is applied to both TiGE-2 and TiGE-3, different indicators and different ranking methods result in different performance.

For the problem with difficulty triplet (0.5, 0.5, 0.5), as shown in Fig. 3(b) and (c), the solutions of TiGE-2 covers all objectives and those of TiGE-3 does not converge well on the seventh and ninth objectives. TiGE-3 performs poorly in terms of maintaining diversity. For the problems with difficulty triplet (0, 0, 0.5) and (0, 0.5, 0), as shown in Fig. 3(e), (f), (h), and (i), the solutions obtained by TiGE-3 concentrate on several parts of the PF, which indicates that TiGE-3 has worse diversity performance than TiGE-2. For the problem with difficulty triplet (0.5, 0, 0), as shown in Fig. 3(l), TiGE-3 cannot cover the region near 0 on all objectives, which means that it has trouble obtaining the boundary solutions on the PF.

3) Summary and Discussions: Table IV gives the final ranking of three TiGE instantiations by the Friedman test on all problems. Obviously, TiGE-2 ranks first, followed by TiGE-1, and TiGE-3. Tables S7–S9, in the supplementary material show that the average feasibility rate of TiGE-2 is 99%, which

TABLE IV
AVERAGE RANKING OF ALL TIGE INSTANTIATIONS BY FRIEDMAN TEST ON ALL TEST INSTANCES

HV	Average ranking value	Final rank
TiGE-2	1.2083	1
TiGE-1	2.2708	2
TiGE-3	2.5208	3

TABLE V
EXISTING INDICATORS AND RANKING METHODS

Ref.	**Indicators for convergence and diversity**	**Ranking or selection**
[39]	$c(X) = agg(f_1(X), \ldots, f_M(X))$ $d(X) = (d_1(X), \ldots, d_{\lvert\mathcal{Q}\rvert}(X))$	One by one selection
[52]	$\lVert F'(X)\rVert = \lVert F(X) - z^*\rVert$ $A(X) = \min_{X \neq Y} \text{angle}(X, Y)$	One by one elimination
[53]	$Cv(X) = 1 - \frac{dis(X)}{\sqrt{M}}$ $Cd(X) = \text{normalization}(SDE(X))$	Scalarization by weighted sum $fit(X) = \alpha \cdot Cd(X) + \beta \cdot Cv(X)$
[54]	$\lVert F'(X)\rVert = \lVert F(X) - z^*\rVert$ $\theta = \arccos \frac{F' \cdot V}{\lVert F'\rVert}$	Scalarization by multiplication $fit(X) = (1 + P(\theta)) \cdot \lVert F'(X)\rVert$

TABLE VI
STATISTICS RESULTS OF TIGE-2 WITH THREE STATE-OF-THE-ART ALGORITHMS IN TERMS OF HV

TiGE-2 vs	$w/t/l$	$R+$	$R-$	p-value	$\alpha = 0.05$
CMOEA/DD	119/9/16	9588.0	852.0	<0.05	YES
CNSGA-III	106/14/24	9486.0	954.0	<0.05	YES
CVaEA	120/10/14	9890.0	550.0	<0.05	YES

TABLE VII
AVERAGE RANKING OF ALL ALGORITHMS BY FRIEDMAN TEST ON ALL TEST INSTANCES

HV	Average ranking value	Final rank
TiGE-2	1.4514	1
CNSGA-III	2.4861	2
CMOEA/DD	2.9861	3
CVaEA	3.0764	4

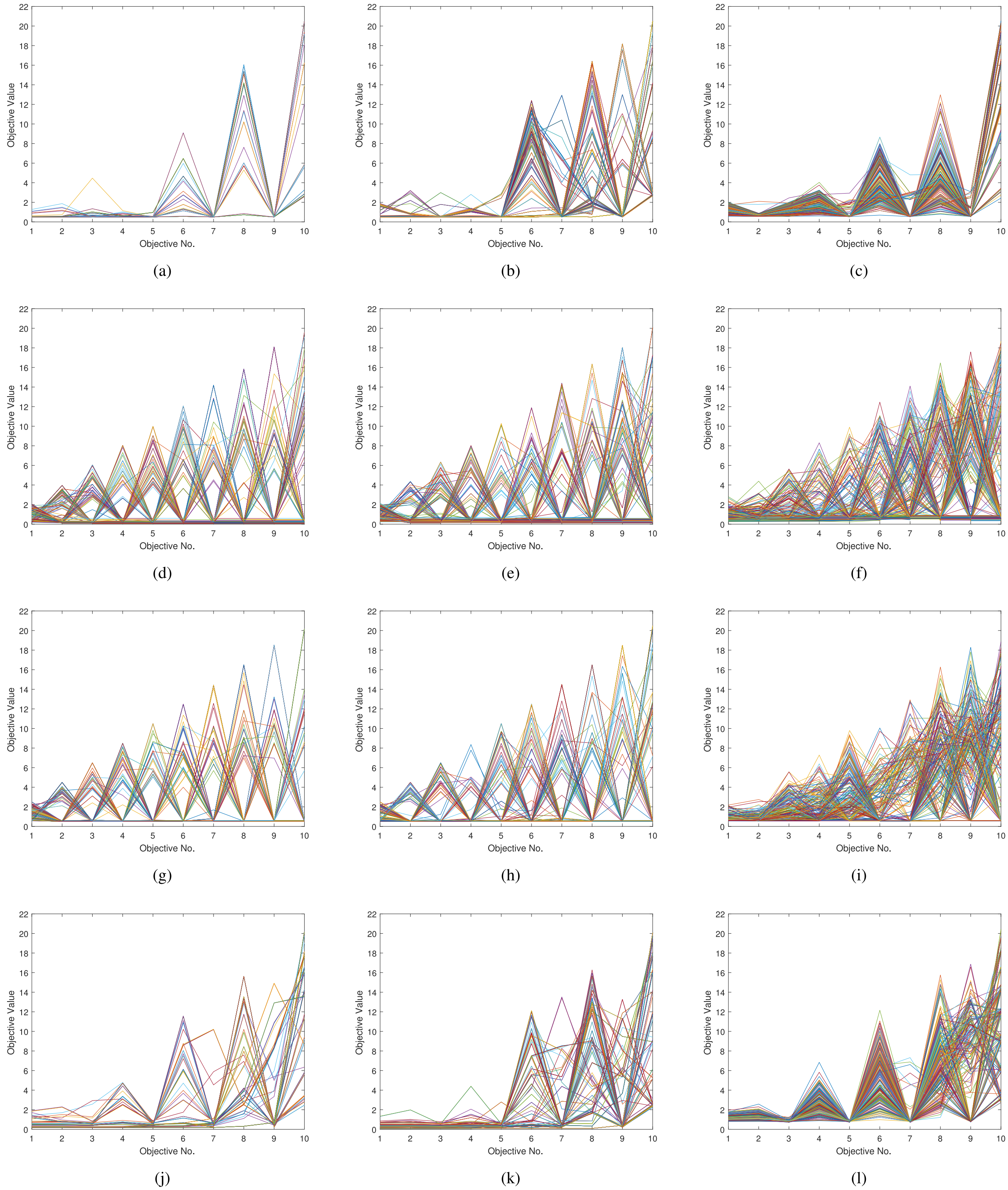


Fig. 3. Approximation set obtained by the TiGE variants on the 10-objective DAC-MaOP4 with different difficulty triplets, shown by parallel coordinates. (a)–(c) $D = (0.5, 0.5, 0.5)$. (d)–(f) $D = (0, 0, 0.5)$. (g)–(i) $D = (0, 0.5, 0)$. (j)–(l) $D = (0.5, 0, 0)$. (a), (d), (g), and (j) obtained by TiGE-1. (b), (e), (h), and (k) obtained by TiGE-2. (c), (f), (i), and (l) obtained by TiGE-3.

is better than TiGE-1 and TiGE-3. Thus, TiGE-2 is selected to compare with the existing state-of-the-art algorithms in the next section.

As described above, TiGE-1 is an extension of BiGE for CMaOPs. It equally treats the three goals with nondominated sorting. Nevertheless, it is unreasonable to always allot the same importance to the three goals because feasibility should take precedence over convergence and diversity. In the one-level balance scheme, simple nondominated sorting of (13) could produce many solutions with good diversity

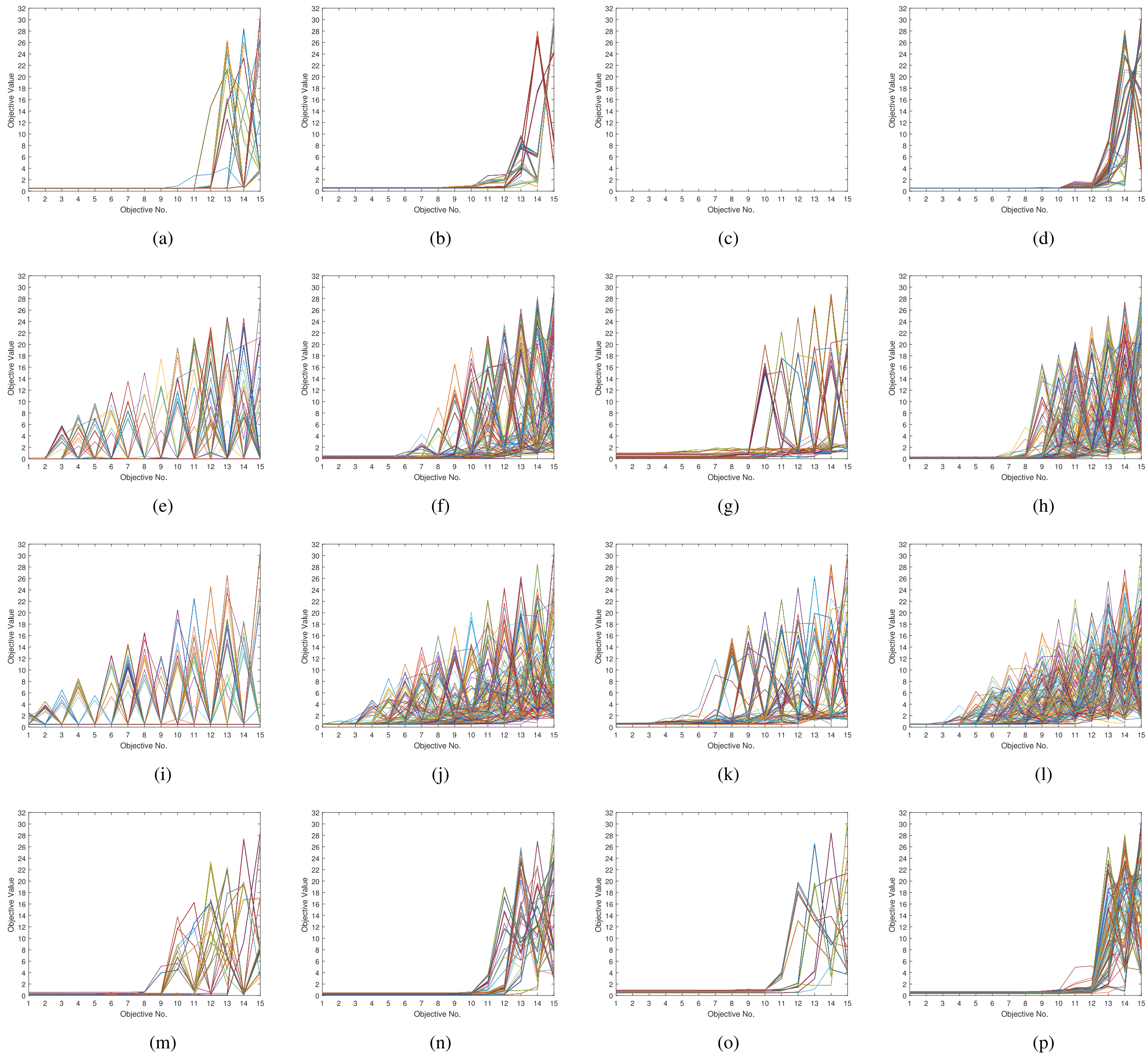


Fig. 4. Approximation set obtained by all algorithms on the 15-objective DAC-MaOP9 with different difficulty triplets, shown by parallel coordinates. (a)–(d) $D = (0.5, 0.5, 0.5)$. (e)–(h) $D = (0, 0, 0.5)$. (i)–(l) $D = (0, 0.5, 0)$. (m)–(p) $D = (0.5, 0, 0)$. (a), (e), (i), and (m) obtained by TiGE-2. (b), (f), (j), and (n) obtained by CNSGA-III. (c), (g), (k), and (o) obtained by CMOEA/DD. (d), (h), (l), and (p) obtained by CVaEA.

but poor feasibility. However, TiGE-2 with a two-level balance scheme deals with the balance between convergence and diversity first, and then uses an adaptive penalty factor to adjust the balance between many-objective optimization (convergence and diversity) and feasibility dynamically.

TiGE-3, as TiGE-2, also adopts the two-level balance scheme, but stochastic ranking is used as the ranking method. TiGE-3 can be seen as the extension of SRA for CMaOPs. Comparisons of TiGE-2 with TiGE-3 show that different indicators and different ranking methods result in different performance since both of them use the two-level balance scheme.

TiGE is just a general framework, and the performance of TiGE is certainly dependent on the components of TiGE. This is the reason that the performance gap among the instantiations of TiGE (TiGE-1, TiGE-2, and TiGE-3) is quite large. In the future, more existing indicators and ranking methods, summarized in Table V, can be further studied and carefully adapted to instantiate the proposed framework. Furthermore, more possible combinations of different components should be adopted, and thus interaction among components in TiGE framework may be further studied.

E. Comparison With State-of-the-Art Algorithms

To further investigate the competitiveness of the proposed framework, three state-of-the-art CMaOEAs, CNSGA-III [3], CMOEA/DD [9], and CVaEA [10], are tested for comparison with TiGE-2. As reviewed in Section II-C, these algorithms are representative CMaOEAs proposed recently. On the one hand, these algorithms are based on powerful unconstrained many-objective optimization algorithms, NSGA-III [23], MOEA/DD [9], and VaEA [38], which deal with the balance between convergence and diversity well. On the other hand, these algorithms adopts different CHTs to utilize the infeasible solutions during evolutionary process, which deal with the balance between objectives and constraints well.

Detailed numerical in terms of HV obtained by the algorithms on five, seven, ten, and fifteen-objective DAC-MaOP problems with different difficulty triplets are summarized in Tables S4–S6, in the supplementary material. Tables VI and VII give the statistics of comparison results between TiGE-2 and three state-of-the-art algorithms. Four different difficulty levels for each problem are tested. As shown in column $w/t/l$ of Table VI, TiGE-2 significantly outperforms the competitors on most problems in terms of HV. Additionally, it is obvious that TiGE-2 obtains higher $R+$ value than $R-$ value and the p value is far less than 0.05. These results indicate that TiGE-2 has great advantage over peer algorithms. Table VII shows the rankings of all algorithms by Friedman test on all instances. As can be seen, TiGE-2 gets the first rank in terms of HV. It can be found that the average feasibility rate of TiGE-2 is better than all competitors from Tables S7–S9, in the supplementary material.

Fig. 4 plots the approximation sets obtained by TiGE-2 and three state-of-the-art CMaOEAs on the 15-objective DAC-MaOP9 with different difficulty triplets by parallel coordinates. For the problem with difficulty triplet (0.5, 0.5, 0.5), as shown in Fig. 4(a)–(d), the solution set of TiGE-2 is distributed more widely than peer algorithms, which indicates the superior performance of TiGE-2 in terms of diversity. Note that nothing is depicted in Fig. 4(c), because CMOEA/DD cannot find any feasible solution in the test problem with this challenging difficulty triplet. Fig. 4(e) clearly demonstrates that TiGE-2 is able to find a well converged and widely distributed set of solutions for the problem with difficulty (0, 0, 0.5). In contrast, the competitors only obtain parts of the true PF, as shown in Fig. 4(f)–(h). Similar observations are found on the problem with difficulty triplet (0, 0.5, 0), as shown in Fig. 4(i)–(l).

The solution sets of the competitors are located in the boundary of the PF on the first objectives as the lines are distributed near 0, which indicates TiGE-2 has better spread over PF than peer algorithms. For the problem with difficulty triplet (0.5, 0, 0), as shown in Fig. 4(m)–(p), the solutions obtained by CNSGA-III, CMOEA/DD, and CVaEA concentrate on 11th to 15th objectives, while TiGE-2 is able to cover certain region on 9th to 15th objectives. Hence, it indicates that TiGE-2 performs better in terms of diversity.

VI. Conclusion

The main challenge of CMaOPs is to deal with balance among convergence, diversity, and feasibility. In this paper, a general framework, TiGE, is proposed for CMaOPs. The proposed framework carefully designs two indicators for convergence and diversity, respectively, and converts the constraints into another indicator for feasibility. A variety of balance schemes and ranking methods are designed to achieve the balance among convergence, diversity, and feasibility. Three instantiations of TiGE framework, called TiGE-1, TiGE-2, and TiGE-3, are then given. Experimental results on the test suite, DAC-MaOP1-9, proposed recently show that TiGE is highly competitive with existing state-of-the-art constrained many-objective evolutionary algorithms for CMaOPs.

As a new CMaOP framework, there are still many interesting issues for future studies in TiGE. First, more options of each algorithmic component can be identified as shown in Table V, and thus more novel and powerful TiGE variants can be instantiated and designed by using different combination of components from the same TiGE framework. Second, this paper mainly focuses on selection operator for CMaOPs. In fact, more powerful reproduction operators can be introduced into TiGE framework. For example, adaptive cross-generation DE operator [55] with good convergence-diversity tradeoff and satisfactory exploration-exploitation balance ability can be considered. At the same time, constraint consensus mutation operator [56] of DE can also be introduced. This operator can be used to directly reduce constraint violations of constrained problems. Finally, coordinated mating and environmental selection strategy [52] can be considered in TiGE framework.

References

[1] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y.-S. Ong, "A new decomposition-based NSGA-II for many-objective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 7, pp. 1191–1210, Jul. 2018.

[2] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2018.2807785.

[3] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[4] J. Wang, W. Ren, Z. Zhang, H. Huang, and Y. Zhou, "A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[5] Y. Zhou and J. Wang, "A local search-based multiobjective optimization algorithm for multiobjective vehicle routing problem with time windows," *IEEE Syst. J.*, vol. 9, no. 3, pp. 1100–1113, Sep. 2015.

[6] J. Wang *et al.*, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2016.

[7] T. Lafetá, M. L. Bueno, C. Brasil, and G. M. Oliveira, "MEANDS: A many-objective evolutionary algorithm based on non-dominated decomposed sets applied to multicast routing," *Appl. Soft Comput.*, vol. 62, pp. 851–866, Jan. 2018.

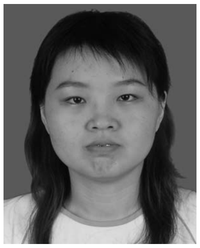
[8] Z. Fan *et al.*, "Difficulty adjustable and scalable constrained multi-objective test problem toolkit," *arXiv preprint arXiv:1612.07603*, 2016.

[9] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[10] Y. Xiang, J. Peng, Y. Zhou, M. Li, and Z. Chen, "An angle based constrained many-objective evolutionary algorithm," *Appl. Intell.*, vol. 47, no. 3, pp. 705–720, Oct. 2017.

[11] H. K. Singh, K. Alam, and T. Ray, "Use of infeasible solutions during constrained evolutionary search: A short survey," in *Proc. 2nd Aust. Conf. Artif. Life Comput. Intell.*, 2016, pp. 193–205.

[12] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, nos. 11–12, pp. 1245–1287, 2002.
[13] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, 2011.
[14] K. Deb, A. Pratap, S. Agarwal, 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.
[15] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control With Applications to Industrial Problems*. Athens, Greece: Int. Center Numer. Methods Eng., 2001, pp. 95–100.
[16] Y. R. Naidu and A. K. Ojha, "Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 6, pp. 820–832, Jun. 2018.
[17] W. Yuan, Y. Liu, H. Wang, and Y. Cao, "A geometric structure-based particle swarm optimization algorithm for multiobjective problems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 9, pp. 2516–2537, Sep. 2017.
[18] 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.
[19] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.
[20] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[21] H.-L. 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.
[22] Q. Kang, X. Song, M. Zhou, and L. Li, "A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2018.2818175.
[23] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
[24] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artif. Intell.*, vol. 228, pp. 45–65, Nov. 2015.
[25] S. Zeng, R. Jiao, C. Li, X. Li, and J. Alkasassbeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2678–2688, Sep. 2017.
[26] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 716–727, Apr. 2015.
[27] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2819208.
[28] W. Gong, Y. Wang, Z. Cai, and S. Yang, "A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 697–713, Oct. 2017.
[29] C. A. C. Coello, "Constraint-handling techniques used with evolutionary algorithms," in *Proc. Genet. Evol. Comput. Conf. Companion*, 2016, pp. 563–587.
[30] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 7, no. 5, pp. 445–455, Oct. 2003.
[31] S. B. Hamida and M. Schoenauer, "An adaptive algorithm for constrained optimization problems," in *Proc. 6th Int. Conf. Parallel Problem Solving Nat.*, 2000, pp. 529–538.
[32] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000.
[33] C. Segura, C. A. C. Coello, G. Miranda, and C. León, "Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization," *Ann. Oper. Res.*, vol. 240, no. 1, pp. 217–250, 2016.
[34] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 233–243, May 2005.
[35] X. Li and G. Zhang, "Biased multiobjective optimization for constrained single-objective evolutionary optimization," in *Proc. 11th World Congr. Intell. Control Autom.*, Jun. 2014, pp. 891–896.
[36] N. Dong and Y. Wang, "An unbiased bi-objective optimization model and algorithm for constrained optimization," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 28, no. 8, 2014, Art. no. 1459008.
[37] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, p. 13, 2015.
[38] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, Feb. 2017.
[39] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689–2702, Sep. 2017.
[40] R. Tanabe and A. Oyama, "A note on constrained multi-objective optimization benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 1127–1134.
[41] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 924–938, Dec. 2016.
[42] L. C. Bezerra, M. López-Ibánez, and T. Stützle, "Automatic component-wise design of multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 403–417, Jun. 2016.
[43] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to non-dominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, Apr. 2015.
[44] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
[45] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.
[46] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in inverted generational distance for triangular linear Pareto front," *IEEE Trans. Evol. Comput.*, to be published, doi: 10.1109/TEVC.2017.2776226.
[47] J. Derrac, S. García, 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. Alcalá-Fdez *et al.*, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, 2009.
[49] R. Storn and K. Price, *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces* (TR: International Computer Science Institute). Berkeley, CA, USA: ICSI, 1995.
[50] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[51] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.
[52] Z. He and G. G. Yen, "Many-objective evolutionary algorithms based on coordinated selection strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220–233, Apr. 2017.
[53] Q. Lin *et al.*, "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.
[54] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
[55] X. Qiu, J.-X. Xu, K. C. Tan, and H. A. Abbass, "Adaptive cross-generation differential evolution operators for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 232–244, Apr. 2016.
[56] N. M. Hamza, D. L. Essam, and R. A. Sarker, "Constraint consensus mutation-based differential evolution for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 447–459, Jun. 2016.



Yalan Zhou received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2008.

In 2008, she joined the Guangdong University of Finance and Economic, Guangzhou, China, where she is currently an Associate Professor. Her current research interest includes artificial intelligence and its applications.

Min Zhu received the B.A. degree in mathematics from Northwestern Polytechnical University, Xi'an, China, in 2015. He is currently pursuing the M.S. degree with Sun Yat-sen University, Guangzhou, China.

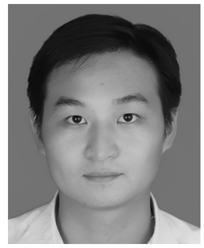
His research interest includes many-objective optimization.

Jiahai Wang (M'07) received the Ph.D. degree in computer science from the University of Toyama, Toyama, Japan, in 2005.

In 2005, he joined Sun Yat-sen University, Guangzhou, China, where he is currently a Professor with the Department of Computer Science. His current research interest includes computational intelligence and its applications.

Zizhen Zhang received the B.S. and M.S. degrees in computer science from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China, in 2007 and 2009, respectively, and the Ph.D. degree in management science from the City University of Hong Kong, Hong Kong, in 2014.

He is currently an Associate Professor with Sun Yat-sen University. His current research interest includes computational intelligence and its applications in production, transportation, and logistics.



Yi Xiang received the B.Sc. and M.Sc. degrees in mathematics from Guangzhou University, Guangzhou, China, in 2010 and 2013, respectively, and the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, in 2018.

He is currently with the School of Software, South China University of Technology, Guangzhou. His current research interests include many-objective optimization and search-based software engineering.

Jun Zhang (M'02–SM'08–F'17) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

From 2004 to 2016, he was a Professor with Sun Yat-sen University, Guangzhou, China. Since 2016, he has been with the South China University of Technology, Guangzhou, where he is currently a Cheung Kong Chair Professor. He has authored seven research books and book chapters, and over 100 technical papers in his research areas. His current research interests include computational intelligence, cloud computing, big data, high performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits.

Prof. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and IEEE TRANSACTIONS ON CYBERNETICS. He is the Founding and Current Chair of the IEEE Guangzhou Section and IEEE Guangzhou Section Computational Intelligence Society Chapters. He is the Founding and Current Chair of the ACM Guangzhou Chapter.