Multifactorial Evolutionary Algorithm Based on Diffusion Gradient Descent

Zhaobo Liu, Guo Li, Haili Zhang, Zhengping Liang, and Zexuan Zhu, *Senior Member, IEEE*

***Abstract*—Multifactorial evolutionary algorithm (MFEA) is one of the most widely used evolutionary multitasking algorithms. MFEA implements knowledge transfer among optimization tasks via crossover and mutation operators, which achieves high-quality solutions more efficiently than the counterpart single-task evolutionary algorithms. MFEA has been successfully applied to various complex optimizations problems, however, there is a lack of convergence proof of the algorithm and the theoretical explanation on how the knowledge transfer can help improve the algorithm performance. To fill this gap, we propose an MFEA based on diffusion gradient descent namely MFEA-DGD in this paper. We prove the convergence of diffusion gradient descent for multiple similar tasks and show that the local convexity of some tasks can help other tasks escape from local optimums by knowledge transfer. On this theoretical foundation, we design new complementary crossover and mutation operators in MFEA-DGD, such that the evolution population has a dynamic equation similar to diffusion gradient descent, i.e., the convergence is guaranteed and the benefit from knowledge transfer is explainable. Specifically, to simulate the principle of gradient descent, the mutation operator is established based on OpenAI evolutionary strategy near the individuals. The crossover operator combines two mutated parents via a generated stochastic matrix to produce offspring. Moreover, to allow MFEA-DGD to explore more undeveloped areas, a hyper-rectangular search strategy based on opposition learning is introduced to search in the uniform search space and the subspace for each task. MFEA-DGD is verified on multi-task optimization benchmarks through a comprehensive empirical study. The experimental results show that MFEA-DGD can convergence faster to competitive results in the comparison with other state-of-the-art evolutionary multitasking algorithms.**

***Index Terms*—Evolutionary multitasking, multifactorial evolutionary algorithm, diffusion gradient descent, convergence analysis, knowledge transfer.**

I. Introduction

Evolutionary multitasking (EMT) [1], [2] represents one of the earliest attempts to solve multiple optimization tasks simultaneously using evolutionary algorithms (EAs). Traditional EAs solve single optimization tasks at a time, but many real-world optimization tasks are related to each other. Valuable knowledge obtained from solving one task can help solve another similar task [3], [4]. EMT has been shown to outperform single-task EAs on various optimization problems, taking advantage of knowledge transfer [5]–[9].

Multifactorial evolutionary algorithm (MFEA) represents the first attempt of EMT and has received fast increasing attention thanks to its simplicity and search abilities [1]. Many MFEA improvements or variants have been proposed in the literature. For example, to reduce the redundant knowledge transfer between low-similar tasks, MFEA was upgraded to MFEA-II [10] by solving the optimal mixture of the component densities based on Kullback-Leibler divergence (KL-divergence). MFEA was further extended to solve many task optimization problems, where more than two tasks are considered [11]. Wen et al. [12] introduced two improvements into MFEA, including the stopping information sharing on the detection of parting ways and reallocating fitness evaluations. In [13], a linear domain adaptive strategy for transforming the search spaces from simple tasks into complex ones was proposed to reduce negative transfer. Feng et al. [14] and Li et al. [15] suggested multi-population evolutionary multitasking patterns and explicit genetic transfer between tasks to facilitate information transfer. Ding et al. [16] proposed a generalized MFEA called G-MFEA to solve computationally expensive problems by simultaneously developing several computationally cheaper assistant problems.

MFEAs have been successfully applied to a variety of complex optimization problems. However, there are very few strict theoretical analyses on the convergence of MFEAs and the benefits of knowledge transfer. Bali et al. [10] presented a proof of algorithm convergence and analyzed the effects of inter-task interactions in MFEA using probability distribution to model the population. However, the convergence of the used probability distribution requires the updating of probability density at each point in the search space, which implies that the entire space must be searched with a population of infinite size. This underlying assumption is usually unrealistic in MFEAs. For other EMT algorithms, Han et al. [17] presented a convergence analysis of a particle swarm optimization based EMT algorithm, which suffers the same issue as [10]. Bali et al. [18] also presented a convergence-guaranteed multi-task gradient descent algorithm (MTGD). The convergence proof holds only when each involved task is convex, whereas many practical optimization problems are non-convex.

To solve the above issues, in this paper we propose a new MFEA based on diffusion gradient descent (DGD) namely MFEA-DGD in which the knowledge transfer and population convergence can be explained in theory. We firstly describe the motivation of using DGD and theoretically prove the

This work was supported in part by the National Natural Science Foundation of China, under Grant 61871272, and the Guangdong Provincial Key Laboratory, under Grant 2020B121201001. (Corresponding author: Zexuan Zhu.)

Z. Liu, G. Li, Z. Liang, and Z. Zhu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China. (e-mail: liuzhaobo@szu.edu.cn, szuliguo@szu.edu.cn, liangzp@szu.edu.cn, zhuzx@szu.edu.cn)

H. Zhang is with the Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen 518055, China. (e-mail:zhanghl@sustech.edu.cn)

validity of knowledge transfer and fast convergence in DGD for multiple similar optimization problems (including non-convex tasks).

Based on DGD, we design new crossover and mutation operators to replace the simulated binary crossover (SBX) [19] and polynomial mutation (PM) [20] in the classical MFEA. With the designed new operators, MFEA-DGD can approximate the dynamic equation of DGD and is endowed with the population convergence and theoretical interpretability of knowledge transfer. Since the analytic form of the gradient of the optimization functions is not available directly, not even exists, we use the estimation method in OpenAI ES [21] to simulate the gradient. Moreover, the hyper-rectangular search strategy inspired by [22] is introduced to MFEA-DGD to enable the exploration of more undeveloped areas. MFEA-DGD is compared with other state-of-the-art MFEAs on benchmark problems and the experimental results demonstrate the efficiency of MFEA-DGD. The main contributions of this work can be summarized as follows:

- The convergence of DGD for multitasking is theoretically proved for the first time in this work. Under some basic assumptions, we prove that DGD is effective for non-convex problems and it can quickly converge near the global optimal solutions by gradient information transfer between tasks.
- By introducing new crossover and mutation operators based on DGD into MFEA, we propose the MFEA-DGD algorithm and enable the algorithm to simulate the optimization process of DGD, which can explain how crossover and mutation improve algorithm performance of similar tasks. This idea can be easily adapted to other types of decentralized optimization algorithms to design EMT algorithms.
- The proposed MFEA-DGD has shown promising performance on benchmark problems. The extensive experimental studies demonstrate to prove that MFEA-DGD can converge very quickly to superior solutions.

The rest of the paper is organized as follows. Section II introduces some preliminaries on MFEA, DGD, and opposition-based learning. Section III presents the theoretical analysis of DGD and the design principles of the operators. Section IV describes the MFEA-DGD method in detail. Section V investigates the performance of the proposed method by empirical experiments. Finally, Section VI concludes this work. For the ease of reference, Table I below provides a summary of the symbols used in this article.

TABLE I: Summary of notation conventions used in the article

$\otimes$	Kronecker product.
tr(A)	Trace of matrix A.
col$\{a, b\}$	Column vector with entries a and b.
diag$\{a, b\}$	Diagonal matrix with entries a and b.
$\lambda_{\min}(A)$	Smallest eigenvalues of matrix A.
$\|x\|$	Euclidean norm of its vector argument.
$\|A\|$	2-induced norm of matrix A (its largest singular value).
$\|A\|_1$	The maximum absolute column sum of matrix A.
A^*	Adjugate of matrix A (transpose of its cofactor matrix).
$A[i, j]$	(i, j)th entry of matrix A.
I_K	Identity matrix of size $K \times K$.

II. PRELIMINARIES

In this section, we present the preliminaries of the involved methodologies in MFEA-DGD to make this paper self-contained. We firstly introduce the conventional MFEA followed by the principle of the DGD algorithm and OpenAI strategies that are the basis for designing the crossover and mutation operators in MFEA-DGD. Lastly, we describe the opposition-based learning that is used in the hyper-rectangular search strategy .

A. Multifactorial Evolutionary Algorithm

Without loss of generality, a multi-task optimization (MTO) problem can be defined as follows:

$$\{\arg\min f_1(\theta_1), \arg\min f_2(\theta_2), \ldots, \arg\min f_n(\theta_n)\} \quad (1)$$

where $\theta_i \in \mathbb{R}^{d_i}$ is the decision variable of the optimization task f_i and $\mathbb{R}^{d_i}$ is the d_i-dimensional search domain. MFEA [1] optimizes the multiple tasks defined in (1) simultaneously in a unified search space with dimension $d = \max d_i$ through a population of individuals. The following properties are defined to quantify the ability of each individual to handle the tasks:

1) Factorial Cost: The factorial cost f_p^i of an individual p is defined as the fitness value of p in terms of a particular task f_i.
2) Factorial Rank: The factorial rank r_p^i of an individual p indicates the rank of p in the population that is sorted in ascending order with respect to task f_i.
3) Skill Factor: The skill factor τ_p of an individual p is the task on which the rank of p is higher than that on the other tasks.
3) Scalar Fitness: The scalar fitness φ_p of an individual p is defined as $\varphi_p = 1/\tau_p$.

Based on the previous definitions, the framework of MFEA is outlined in Algorithm 1. In the initial stage of the algorithm, the population consists of N individuals randomly generated in a unified express space, then each individual is randomly assigned a skill factor and evaluated in terms of factorial cost. Afterward, in each generation of evolution, assortative mating and vertical cultural transmission mechanism are applied to reproduce offspring through crossover and mutation operators. The knowledge between different tasks is shared by exchanging genetic information between individuals. The vertical cultural transmission mechanism enables individuals with different skill factors to mate with a certain probability. The optimization of each task benefits from other tasks through this mechanism. Once the offspring population is generated, the factorial cost, factorial rank, scalar fitness, and skill factor of each individual are updated. Elite-based environmental selection is then applied to generate a new population from the union of the parent and offspring populations. The above evolution procedure repeats until some stopping criterion is reached.

Algorithm 1 The Framework of MFEA

```
Input: $N$ (population size), $n$ (number of tasks)
Output: a series of solutions
Initialize the population $P$
Randomly assign the skill factor for each individual in $P$
Evaluate factorial cost of each individual
while the stopping criteria are not reached do
    Generate offspring population $O$ based on assortative mating
    Perform vertical cultural transmission
    Evaluate offspring individuals
    Generate new population $P' = P \cup O$
    Update the scalar fitness $\varphi$ and skill factor $\tau$ of each individual
    Select the $N$ fittest individuals from $P'$ to form $P$
end while
```

Algorithm 2 Diffusion Gradient Descent (DGD)

```
Input: $\theta_{0,i}$, $i = 1, \ldots, n$, step size $\eta$, matrix $\mathcal{A} = \{a_{ij}\} \in \mathbb{R}^{n \times n}$
  for $t = 0, 1, \ldots,$ do
    $\theta_{t+1,i} \leftarrow \sum_{j=1}^{n} a_{ij}(\theta_{t,j} - \eta \nabla f_j(\theta_{t,j}))$, $i = 1, \ldots, n$
  end for
```

B. Diffusion gradient descent

Given an n-task optimization problem with each task $i \in \{1, \ldots, n\}$ aiming to solve the corresponding optimization problem,

$$\min_{\theta_i} f_i(\theta_i) \tag{2}$$

where f_i can be non-convex. Without out loss of generality, we suppose $\theta_i \in \mathbb{R}^d$. We begin by considering the case in which exact gradients are available, such that gradient descent (GD) can be implemented. At time t, each task i derives a candidate solution $\theta_{t,i}$ and a gradient information $\nabla f_i(\theta_{t,i}) \in \mathbb{R}^d$. For convex problems, GD is efficient, but in a non-convex problem, GD algorithm tends to stuck at local optimums. To escape local optimums by using useful information between similar tasks, we consider diffusion strategies [23] on GD. The diffusion strategies can be beneficial compared to purely non-cooperative strategies provided that the local optimums are sufficiently close to each other [24].

A typical version of DGD used in this paper is presented in Algorithm 2. Let $\theta_{t,i}$ denote the estimate of the minimizer of task i and time instant t. Similar to the diffusion LMS [24], the general structure of DGD algorithm consists of the following steps:

$$\left\{ \begin{array}{l} \phi_{t+1,i} = \sum_{l=1}^{n} a_{1,li}\theta_{t,l} \\ \varphi_{t+1,i} = \phi_{t+1,i} - \eta \sum_{l=1}^{n} c_{li} \nabla f_l(\phi_{t,l}) \\ \theta_{t+1,i} = \sum_{l=1}^{n} a_{2,li}\varphi_{t+1,l}. \end{array} \right. . \tag{3}$$

The non-negative coefficients $a_{1,li}$, $a_{2,li}$ and c_{li} are the (l,i)-th entries of two left-stochastic matrices, $\mathcal{A}_1$ and $\mathcal{A}_2$, and a right-stochastic matrix $\mathcal{C}$, i.e.,

$$\mathcal{A}_1^T \mathbf{1}_n = \mathbf{1}_n, \quad \mathcal{A}_2^T \mathbf{1}_n = \mathbf{1}_n, \quad \mathcal{C}\mathbf{1}_n = \mathbf{1}_n, \tag{4}$$

Several adaptive strategies can be obtained as special cases of (3) through appropriate selections of $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{C}$. For instance, the setting $\mathcal{A}_1 = I_n$ yields the so-called adapt-then-combine (ATC) DGD. The setting $\mathcal{A}_2 = I_n$ leads to the combine-then-adapt (CTA) DGD. By setting $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{C} = I_n$, the algorithm degenerates to non-cooperative standard gradient descent. According to [23], the ATC diffusion LMS algorithm tends to outperform the CTA version, so we adopt the ATC version of DGD in this paper. To facilitate the follow-up study, we consider a common case [25], [26] with $\mathcal{C} = I_n$, where (3) becomes

$$\theta_{t+1,i} = \sum_{j=1}^{n} a_{ij}(\theta_{t,j} - \eta \nabla f_j(\theta_{t,j})), \quad i = 1, \ldots, n, \tag{5}$$

and $\mathcal{A}_2 = \mathcal{A} = \{a_{ij}\}_{n \times n}$.

C. Open AI evolutionary strategy

Assume that $f(\theta)$ is only available by virtue of function evaluations, and the gradient $\nabla f(\theta)$ is inaccessible. We introduce the notion of Gaussian smoothing [27] of the objective function $f(\theta)$. Let $\sigma > 0$ be the smoothing parameter and $f_\sigma(\theta)$ denote the Gaussian smoothing of f with radius σ, i.e.,

$$\begin{aligned} f_\sigma(\theta) &= E_{\varepsilon \sim N(0,I_d)} f(\theta + \sigma\epsilon) \\ &= \frac{1}{\pi^{d/2}} \int_{\theta \in \mathbb{R}^d} f(\theta + \sigma\epsilon) e^{-\frac{\|\epsilon\|^2}{2}} d\epsilon. \end{aligned}$$

We remark that $f_\sigma(\theta)$ preserves important features of the objective function including, e.g., convexity and the Lipschitz constant, and is always differential even when $f(\theta)$ is not. The gradient of $f_\sigma(\theta)$ can be computed as

$$\nabla f_\sigma(\theta) = \frac{2}{\sigma} E_{\varepsilon \sim N(0,I_d)} \varepsilon f(\theta + \sigma\varepsilon). \tag{6}$$

Then for $M \in \mathbb{N}^+$, $\{\varepsilon_j\}_{j=1}^{M}$ are independent and identically distributed, $\varepsilon_1 \sim N(0, I_d)$. Traditional gradient-free optimization (GFO) methods estimate (6) via Monte Carlo (MC) sampling and provide an iterative update to the state θ, given by [21]

$$\nabla f_\sigma(\theta) \approx \frac{2}{\sigma M} \sum_{m=1}^{M} \varepsilon_m f(\theta + \sigma\varepsilon_m). \tag{7}$$

The primary advantage of such GFO approaches is that they are easy to implement, embarrassingly parallelizable, and can be easily scaled to include a large number of workers. A promising attempt to improve the efficiency and accuracy of the MC gradient estimate (7) is to consider decoupling the problem (6) along d orthogonal directions [28]. The gradient can be estimated by virtue of, e.g., an antithetic orthogonal sampling, i.e.,

$$\nabla f_\sigma(\theta) \approx \frac{1}{\sigma M} \sum_{j=1}^{M} (\epsilon_j f(\theta + \sigma\epsilon_j) - \epsilon_j f(\theta - \sigma\epsilon_j)). \tag{8}$$

where $\{\epsilon_j\}_{j=1}^{M}$ are marginally distributed as $N(0, I_d)$, and the joint distribution of $\{\epsilon_j\}_{j=1}^{M}$ is defined as follows. If $M \leq d$, then the vectors are conditioned to be orthogonal almost surely. If $M > d$, then each consecutive set of d vectors is conditioned to be orthogonal almost surely, with distinct sets of d vectors remaining independent. Using the orthogonal directions, as opposed to the MC directions defined in (7), improves the overall performance when approximating (6). Therefore, for a given individual and its corresponding skill factor, we can use additional $2M$ evaluations to find an evolutionary direction and obtain its quasi-gradient.

D. Opposition-based learning

OBL was introduced in [29] as a new scheme for machine intelligence and has been successfully applied to various population-based evolutionary algorithms [30], [31]. A population of individuals randomly generating offspring near itself tends to repeatedly access the areas where there is no global optimal solution [32]. To overcome this problem, OBL considers both candidate individuals and their corresponding opposite individuals in the search space, thus well expanding the search range. In this paper, we use the generalized concept of OBL in [22] to solve MTO problems. The basic definitions of opposite solution are presented as follows:

Opposite Point: given a real number γ in range $[a, b]$, the opposite point of γ denoted as $\overline{\gamma}$ is defined as

$$\overline{\gamma} = a + b - \gamma \tag{9}$$

d-dimensional Opposite Point: the concept of opposite point can be generalized to d-dimensional space, where $d \geq 2$. Given an vector $\Gamma = \{\gamma_1, \dots, \gamma_d\}$ and two boundary vectors $A = \{a_1, \dots, a_d\}$ and $B = \{b_1, \dots, b_d\}$ with $\gamma_i \in [a_i, b_i]$, where $i = 1, \dots, d$. The opposite vector, or d-dimensional opposite point, of Γ is defined as $\overline{\Gamma} = (\overline{\gamma}_1, \dots, \overline{\gamma}_d)$ with

$$\overline{\gamma}_i = a_i + b_i - \gamma_i. \tag{10}$$

III. Convergence proof of DGD and its application in operators design

In this section, we will first introduce a new theoretical analysis method to show that the DGD is suitable for non-convex optimization problems and that the gradient information of each task can be combined effectively to achieve fast convergence. Next, we introduce the idea of the proposed new crossover and mutation operators inspired by DGD with the gradient simulated by the OpenAI evolutionary strategy.

A. Convergence of DGD

Different from the existing research, here we show that the convergence of Algorithm 2 does not require any convexity condition of each f_i in essence, and the key to the convergence of DGD lies in the strong-convexity of $f^{glob} \triangleq \sum_{i=1}^{n} f_i$. Before describing the theorem, we give several definitions:

Definition 1. *An square matrix M is said to be row-allowable if it has at least one positive entry in each row. A row-allowable matrix is called scrambling if any two rows have at least one positive element in a coincident position, i.e., for $M = \{m_{ij}\}$,*

$$\tau_1(M) = 1 - \min_{i,j} \sum_k \min\{m_{ik}, m_{jk}\} < 1.$$

Definition 2. *A matrix $M \in \mathbb{R}^{n \times n}$ is said to be irreducible, there is a sequence $i_1, \dots, i_l$ contains $\{1, \dots, n\}$, satisfies $m_{i_k i_{k+1}} > 0$, here $i_{l+1} = i_1$.*

Definition 3. *A differentiable function f is ℓ-**gradient Lipschitz** if:*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq \ell \|x_1 - x_2\| \quad \forall\, x_1, x_2.$$

Definition 4. *A twice-differentiable function f is ρ-**Hessian Lipschitz** if:*

$$\|\nabla^2 f(x_1) - \nabla^2 f(x_2)\| \leq \rho \|x_1 - x_2\| \quad \forall\, x_1, x_2.$$

Next, we analyze the asymptotic properties of $\{\theta_{t,i}\}$ under

Assumption 1. *The functions $\{f_i, i = 1, \dots, n\}$ are ℓ-gradient Lipschitz and ρ-Hessian Lipschitz.*

Assumption 2. *There exists positive constant ξ such that $\nabla^2 f^{glob} \geq \xi I_d$.*

Assumption 3. *$\mathcal{A}$ is a scrambling doubly stochastic matrix with $\mathcal{A}^\tau \mathcal{A}$ being irreducible.*

For fixed $\theta \in \mathbb{R}^d$, we denote $\tilde{\theta}_{t,i} \triangleq \theta_{t,i} - \theta$, then

$$\widetilde{\Theta}_{t+1} = (\mathcal{A} \otimes I_d)(I_{dn} - F_t)\widetilde{\Theta}_t - (\mathcal{A} \otimes I_d) L_t,$$

where

$$\begin{aligned}
\widetilde{\Theta}_t &\triangleq \mathrm{col}\{\tilde{\theta}_{t,1}, \dots, \tilde{\theta}_{t,n}\}, \\
L_t &\triangleq \eta \cdot \mathrm{col}\{\nabla f_1(\theta), \dots, \nabla f_n(\theta)\}, \\
F_t &\triangleq \eta \cdot \mathrm{diag}\{H_{t,1}, \dots, H_{t,n}\}, \\
H_{t,i} &\triangleq \int_0^1 \nabla^2 f_i(\theta + \eta(\theta_{t,i} - \theta)) d\mu. \quad i = 1, \dots, n.
\end{aligned}$$

Theorem 1. *Under Assumption 1–3, for any $r > 0$ and $\theta \in \mathbb{R}^d$, there is $\eta^* > 0$ such that for any $\eta \in (0, \eta^*)$ and initial value $\theta_{0,1} = \ldots = \theta_{0,n}$ with $\|\theta_{0,1} - \theta\| \leq r$,*

$$\limsup_{t \to +\infty} \|\widetilde{\Theta}_t\| \leq \frac{\|\nabla f^{glob}(\theta)\|}{s\xi}(2 + \sqrt{3 + \eta s \xi}) \tag{11}$$

here s is a constant depends on $\mathcal{A}$. The proof of this theorem is provided in the Appendix.

Remark 1. *Define $\widehat{\Theta}_t = col\{\theta_{t,1} - \theta_1^*, \dots, \theta_{t,n} - \theta_n^*\}$, where $\theta_i^* = \underset{\theta}{argmin}\, f_i(\theta)$. Choose $0 < \eta < \min\{(s\xi)^{-1}, \eta^*\}$, (11) implies*

$$\limsup_{t \to +\infty} \|\widehat{\Theta}_t\| \leq \frac{4\|\nabla f^{glob}(\theta)\|}{s\xi} + \sqrt{\sum_{i=1}^{n} \|\theta - \theta_i^*\|^2} \triangleq G(\theta),$$

so $\limsup_{t \to +\infty} \|\widehat{\Theta}_t\| \leq \inf_\theta G(\theta)$. We assert that for similar tasks $f_1, \dots, f_n$, $\inf_\theta G(\theta)$ must be small. Actually, by a simple calculation we have

$$\inf_\theta G(\theta) \leq G(\theta_1^*) \leq \left(\frac{4\ell}{s\xi} + 1\right) \sqrt{\sum_{i=2}^{n} \|\theta_1^* - \theta_i^*\|^2}, \tag{12}$$

so, similarity between n tasks can lead a satisfactory approximation of optimal solutions $\{\theta_i^*\}_{i=1}^n$ under evolutionary strategy (5).

Remark 2. *Observing Assumption 2, we do not need the convexity of each optimization task. For example, even if $f_1, \dots, f_{n-1}$ are non-convex functions, as long as the convexity of f_n is enough to ensure the strong-convexity of f^{glob}, the algorithm can still converge near the optimal solution. Therefore, as long as there is one task with good convexity in a group of similar tasks, it can provide more gradient information to help other tasks approach their respective optimal solutions.*

Remark 3. *It is worth mentioning that matrix $\mathcal{A}$ in algorithm 1 can be replaced by the time-varying $N \times N$ matrix $\mathcal{A}_t = \{a_{t,li}\}$, the gradient of functions $\nabla f_1, \dots, \nabla f_n$ can also be replace the time-varying function group $\nabla f_{t,1}, \dots, \nabla f_{t,N}$, where each $f_{t,i} \in \{f_1, \dots, f_n\}$, which is of great significance to the theoretical interpretation of the evolutionary algorithm we designed in the later sections. At this case, conclusion similar to Theorem 1 can still be established, if we assume that there is integer $L > 0$ such that $\sum_{j=t}^{t+L} \sum_{i=1}^{N} \nabla^2 f_{j,i} \geq \xi I_d$, each $B_t = A_{t+L}A_{t+L-1} \cdots A_t$ satisfies Assumption 3 and the non-zero entries of the matrices $\mathcal{A}_t$ have the following uniform lower bound a:*

$$\min_{(l,i):a_{t,li}>0} a_{t,li} \geq a > 0, \quad t > 0. \tag{13}$$

Since the proof process is almost the same, we omit it in this paper. In addition, readers interested in this can refer to Assumption 6 in [33], which is very similar to our Assumptions here.

B. New crossover and mutation operators

Since DGD has excellent convergence and interpretability, this inspires us to design new crossover and mutation operators based on DGD. Generally, crossover denotes the reproduction operator for exchanging genetic materials between parents and generating offspring in evolutionary algorithms. In the last few decades, a large number of crossover operators have been proposed in the literature for a wide range of optimization problems [34]–[36]. We first focus on two kinds of typical crossover operators, Arithmetical and SBX, where the element at position i of the offspring is the linear combination of the two selected parents:

$$\begin{cases} c_1^i = \nu \cdot p_1^i + (1-\nu) \cdot p_2^i \\ c_2^i = (1-\nu) \cdot p_1^i + \nu \cdot p_2^i \end{cases}. \tag{14}$$

The only difference between these two crossover opterators is whether ν is selected randomly. The above formula can be rewritten in the form of matrix multiplication:

$$\mathbf{c} = \mathcal{A} \cdot \mathbf{p}. \tag{15}$$

If the population size $N = 2$, we iteratively generate new offspring according to such a crossover operator. Consider an ideal case that the offspring generated in each step have better fitness than their parents, then the population of generation t can by represented as

$$\mathbf{p}_t = \mathcal{A}_{t-1} \cdot \mathcal{A}_{t-2} \cdots \mathcal{A}_0 \cdot \mathbf{p}_0. \tag{16}$$

Such a formula cannot guarantee the convergence rate of the population to the optimal solutions, but only makes linear transformations between individuals. Inspired by the DGD algorithm and Theorem 1, we make the following modification to the above crossover operator (15):

$$\mathbf{c} = \mathcal{A} \cdot \mathbf{p}' \approx \mathcal{A}(\mathbf{p} - \eta \nabla \mathbf{f}(\mathbf{p})), \tag{17}$$

where $\mathbf{f}$ is a two-dimensional vector valued function, of which each component belongs to the set $\{f_1, \dots, f_n\}$, and the gradient $\nabla \mathbf{f}$ can be approximated by OpenAI evolutionary strategy referred in Section II-C.

For individual mutation, in the literature, PM is one of the most commonly used mutation operators, which works by randomly mutating parents to produce offspring. Nevertheless, we expect the mutation to have a certain directionality because natural evolution is influenced by environmental influences and is not entirely random. Parents need to get some empirical information from other individuals for their mutations. Theoretically, we designed the mutation operator based on GD. We expect the offspring $\mathbf{c}$ to be generated in a quasi-gradient descent direction,

$$\mathbf{c} \approx \mathbf{p} - \eta \nabla \mathbf{f}(\mathbf{p}), \tag{18}$$

which is a special case of (17) with $\mathcal{A} = I_2$.

We apply the crossover (17) and mutation (18) to a population P_{t-1} of N individuals. Assuming that in the given $N/2$ pair of parents, $N_1/2$ pairs use crossover operators and the remaining $(N - N_1)/2$ pairs use mutation operators. The iteration of the whole population must satisfy the following equation

$$P_t \approx \mathcal{A}_{t-1}(P_{t-1} - \eta \nabla \mathbf{f}_{t-1}(P_{t-1})), \tag{19}$$

where

$$\begin{aligned} \mathcal{A}_{t-1} &= diag\{A_{t-1,1}, \dots, A_{t-1,N_1/2}, \underbrace{I_2, \dots, I_2}_{(N-N_1)/2}\}, \\ \mathbf{f}_{t-1} &= (f_{t-1,1}, \dots, f_{t-1,N})^T, \end{aligned} \tag{20}$$

and $A_{t-1,i} \in \mathbb{R}^{2\times 2}$, $i = 1, \dots, N/2$, $f_{t-1,i} \in \{f_1, \dots, f_n\}$, $i = 1, \dots, N$. Equation (19) is completely consistent with the recursive equation of the DGD algorithm described in Section II-B. Therefore, if the crossover operator and mutation operator are designed according to (17) and (18), respectively, the advantage of DGD can be retained by choosing proper $\mathcal{A}_i$, $i \geq 0$ (as described in Remark 3). Compared with the traditional operators, such a strategy has a faster convergence rate and can overcome the optimization of non-convex tasks. Moreover, the knowledge transfer and the convergence of solutions caused by crossover and mutation operators can be explained theoretically.

IV. Proposed MFEA-DGD

A. Overall framework

Based on the theoretical analysis in the previous section, we propose MFEA-DGD based on the new crossover and mutation operators that enable the algorithm to simulate the optimization process of DGD. The pseudo code of MFEA-DGD is summarized in Algorithm 3. Generally, the main difference between MFEA-DGD and the conventional MFEA

Algorithm 3 MFEA-DGD

```
Input: $N$ (population size), $n$ (number of tasks), $M$(number of individuals to simulate the gradient), $\sigma$(smoothing parameter)
Output: a series of solutions
Initialize population $P$; Randomly assign skill factor $\tau$ for every individual; Initialize quasi-Lipschitz constant $L$
while not reach maximum fitness evaluation do
    for $i = 1, 2 \ldots, N/2$ do
        Let learning late $\eta = \sigma/L$
        Randomly select two parent individuals $p_1, p_2$
        Randomly generate a matrix $\mathcal{A} \in \mathbb{R}^{2\times 2}$
        Obtain skill factor $\tau_1, \tau_2$ of $p_1, p_2$, respectively
        Let $\{\xi_j^i\}_{j=1}^M$ be marginally distributed as $N(0, I_d)$, $i = 1, 2$
        Obtain individuals $p_{i,-1}^j = p_i - \sigma\xi_j^i$ and $p_{i,1}^j = p_i + \sigma\xi_j^i$ for each $p_i$, $j = 1, \ldots, M$
        if $\tau_1 = \tau_2$ or $rand < rmp$ then
            $o_1 = GradTransform(p_1, p_2, \xi^1, \xi^2, \mathcal{A})$
            $o_2 = Hyper - rectangleSearch(o_1)$
        else
            for $i = 1, 2$ do
                $o_i = Quasi - GradMutation(p_i, \xi^i)$
            end for
        end if
        for $k = 1, \ldots, M$ do
            $o_{4k-1} = p_{1,-1}^k$, $o_{4k} = p_{1,1}^k$, $o_{4k+1} = p_{2,-1}^k$, $o_{4k+2} = p_{2,1}^k$
        end for
    end for
    Evaluate offspring population $O$
    New population $NP = P \cup O$
    Select fittest individuals from $NP$ to form $P$
    Update learning late $\eta$
end while
```

Algorithm 4 The quasi-gradient mutation strategy

```
Input: individual $p$, skill factor $\tau$, matrix $[\xi_1, \ldots, \xi_M] \in \mathbb{R}^{d\times M}$, smoothing parameter $\sigma$, learning rate $\eta$
Output: the generated child $o$
$\nabla f_\sigma(p) = \sum_{j=1}^{M} \frac{\xi_j f_\tau(p+\sigma\xi_j) - \xi_j f_\tau(p-\sigma\xi_j)}{\sigma}$
$o = p - \eta\nabla f_\sigma(p)$
```

algorithms lies in the reproduction operators and the hyper-rectangle search strategy. As shown in Algorithm 3, the workflow of MFEA-DGD can be outlined as follows:

1) At the beginning, MFEA-DGD performs the same initialization as MFEA does to generate a population.
2) In each evolutionary generation, two parent individuals, denoted as p_1 and p_2, are randomly selected. For each p_i, $2M$ candidate offspring $\{p_{i,-1}^k\}_{k=1}^M$ and $\{p_{i,1}^k\}_{k=1}^M$ are randomly generated to simulate the direction of the gradient descent of f_{τ_i} at p_i.
3) The selected parent individuals p_1 and p_2 mate following the assorting mating mechanism that includes the gradient transform strategy. If p_1 and p_2 share the same skill factor, by applying gradient transform and hyper-rectangle search strategies, they generate two offspring individuals o_1 and o_2, respectively. When p_1 and p_2 have different skill factors, they still have a random mating probability (rmp) to activate the two strategies. Otherwise, they generate offspring individuals o_1 and o_2 via quasi-gradient descent mutation operator, respectively. Here rmp is used to adjust the information exchange frequency between different tasks. In the case the similarity between the given n tasks is relatively high, we tend to use a larger value of rmp, i.e., closer to 1. If the similarity between tasks is low, we prefer to use the mutation operator to simulate gradient descent's optimization process directly. Therefore, the corresponding rmp is small.
4) Lastly, after generating and evaluating the offspring population, the elite-based environmental selection operator is applied to form the next generation population. The learning rate(or step size) η is also updated, which can be used to improve the performance of the designed operator.

The critical parts of MFEA-DGD including the gradient transform and quasi-gradient descent mutation search strategies, the update criteria for learning rate $\eta = \sigma/L$, and the hyper-rectangle search, are described in detail in the following subsections.

B. Quasi-gradient descent mutation

Unlike traditional mutation, we design mutation criteria by using an extra number of function evaluations to approximate the gradient descent. This can be understood as individuals choosing the right evolutionary direction based on the experiences of randomly selected individuals around them, consistent with our intuitive understanding of evolutionary processes. The pseudo code is provided in Algorithm 4.

In Algorithm 4, $\{\epsilon_j\}_{j=1}^M$ are marginally generated by standard Gaussian distribution $N(0, I_d)$, which guarantees the randomness of the direction of mutation. $\nabla f_\sigma(p)$ represents the degree of the mutation of p. When there is a large difference in fitness between two individuals $p + \sigma\xi_j$ and $p - \sigma\xi_j$, it is shown that the direction of mutation is likely to decrease the fitness of p more quickly, with the individual p gaining more empirical information from randomly generated individuals $p + \sigma\xi_j$ and $p - \sigma\xi_j$. Conversely, if the fitness gap between two individuals is small, the mutation is relatively small.

C. Gradient transform strategy

During the evolutionary process, most of the offspring solutions are generated by parents specific to different tasks. If n tasks are largely uncorrelated, the generated offspring solutions cannot fit well to either task, i.e., the new solutions struggle to survive to the next generation. As such, the knowledge transfer among tasks becomes less efficient as the evolution goes on. To address this issue, a gradient transform strategy is proposed

Algorithm 5 The gradient transform strategy

```
Input: Individuals $p_1, p_2$, matrices $\xi^1 = [\xi_1^1, \ldots, \xi_M^1], \xi^2 = [\xi_1^2, \ldots, \xi_M^2] \in \mathbb{R}^{d \times M}$, smoothing parameter $\sigma$, learning rate $\eta$, matrix $\mathcal{A} \in \mathbb{R}^{2 \times 2}$
Output: the generated child $o$
for $i = 1, 2$ do
    $\nabla f_{\sigma,i}(p_i) = \sum_{j=1}^{M} \frac{\xi_i^j f_{\tau_i}(p_i + \sigma\xi_i^j) - \xi_i^j f_{\tau_i}(p_i - \sigma\xi_i^j)}{\sigma}$
end for
$o = a_{11}(p_1 - \eta\nabla f_{\sigma,1}(p_1)) + a_{12}(p_i - \eta\nabla f_{\sigma,2}(p_2))$
```

to minimize the sum of similar different tasks and the pseudo code is provided in Algorithm 5. Given two parent individuals p_1 and p_2, and their respective randomly generated quasi-gradient descent offspring, the crossover of the two parents is directly undertaken to generate an offspring. If their skill factors are the same or their most effective tasks are relevant, the resulting offspring might have better fitness. Otherwise, the locations of the two parents could be far away from each other in the unified express space. The crossover of them can increase the diversity of the population. The mapping process is also called gradient transform in this study.

We take the optimization of two tasks T_1 and T_2 as an example to illustrate the mapping process. Firstly, the population P is divided into two sub-populations based on the skill factor of each individual, denoted as pop_1 and pop_2. Assume that $p_1 \in pop_1$ and $p_2 \in pop_2$, the strategy essentially mimics the process of applying gradient descent to each p_i for its corresponding functions, and then makes a linear combination of the resulting solutions p'_1 and p'_2. According to the convexity of f_1 and f_2, there are two cases:

1) $f_1 + f_2$ has good convex properties at the corresponding parent p_1 or p_2, then by Theorem 1, the linear combination of p'_1 and p'_2 is a high-quality offspring and approaches the global optimal solution argmin $f_1 + f_2$ with the rate of gradient descent. Because of the similarity between f_1 and f_2, the global optimal solution is likely to be near the optimal solution for each task. Compared with traditional crossover operators, the gradient transform strategy has a theoretical guarantee and faster convergence speed in this case.
2) $f_1 + f_2$ is poorly convex or even non-convex. At this point, p'_1 or p'_2 from the simulated gradient descent algorithm may fall near the local optimums of f_1 or f_2, so that the offspring o is defined by the linear combination of p'_1 and p'_2, which can helps the offspring escape the local optimums of f_1 and f_2, and avoid obtaining local optimal solutions.

Essentially, the strategy we propose is to communicate gradient information between similar tasks so that fast gradient descent can help those with slow gradient descent escape local optimums effectively and approach global optimal solutions.

D. Learning rate

We consider the adaptive selection of the learning rate η. Instead of using a fixed value for the learning rate or a predetermined schedule, the geometry of the target function is used to derive the step size. For each direction ξ_i^j the values $\{f_{\sigma,\tau_i}(p_i + \sigma\xi_i^j), f_{\sigma,\tau_i}(p_i - \sigma\xi_i^j)\}$, are used to estimate the directional local Lipschitz constants

$$L_i^j = \left| \frac{f_{\sigma,\tau_i}(p_i + \sigma\xi_i^j) - f_{\sigma,\tau_i}(p_i - \sigma\xi_i^j)}{2\sigma} \right|, \tag{21}$$

let $L = \max_{j \in [1,M], i=1,2} L_i^j$, the learning rate η is derived from the smoothing parameter σ and a running average over Lipschitz constant L computed on previous iterations, denoted L_D, i.e.,

$$L_D \longleftarrow (1 - \gamma)L + \gamma L_D, \quad \eta = \sigma / L_D, \tag{22}$$

where $\gamma \in (0, 1)$ is a tunable parameter. As each generation of population is renewed, we get a new η. The update criteria here mainly take into account that the optimal learning rate for gradient descent is generally related to the Lipschitz constant of the function.

E. Hyper-rectangle search strategy

The traditional SBX, PM, and the new operators designed in this paper are intended to reproduce offspring near the parents with high probability, which may limit the whole search range of the population. Therefore, we introduce the rectangle search strategy based on OBL to MFEA-DGD. The pseudo code is summarized in Algorithm 6. It contains two modes. The first is the search in the unified express space, and the second is the search in each local space of the sub-task. In the first mode, we set the upper bound $\mathcal{U}$ and lower bound $\mathcal{L}$ of the unified express space as the search boundary. For a generated offspring o_1, the algorithm retrieves the opposite solution o_2 based on the OBL method as follows,

$$o_2 = \mathcal{U} + \mathcal{L} - o_1. \tag{23}$$

In the second model, the population is divided into n subgroups according to the skill factors, and the upper and lower bounds of the kth-task are configured as $\mathcal{U}_k$ and $\mathcal{L}_k$, respectively. Moreover, a random number sr is used to control the expanding or narrowing of the search range. Specifically, the larger the range of sr, the sub-population can expand the exploration range in more potential areas. The smaller the range of sr, the more the search is focused on the local subspace. The opposite solution o_2 is defined as follows:

$$o_2 = \mathcal{U}_k + \mathcal{L}_k - o_1. \tag{24}$$

The above two modes are applied alternately to balance exploration and exploitation in the unified search space and the sub-spaces.

V. Experiments

On two MTO test suits, we compare the performance of the proposed MFEA-DGD algorithm to that of several state-of-the-art EMT algorithms as well as the single-task optimization counterpart. The MFEA-DGD convergence rate is also being investigated. All experiments are carried out on a PC running Windows, with an Intel Core i7-8700 CPU running at 3.20GHz and 16GB of RAM.

Algorithm 6 The Hyper-rectangle search strategy

```
Input: the generated child $o_1$, The current generation number $t$, The upper and lower boundaries of the $k$-th task $\mathcal{U}_k$, $\mathcal{L}_k$, The upper and lower boundaries of the unified express space $\mathcal{U}$, $\mathcal{L}$
Output: the generated child $o_2$
Generate a random number $sr$ within a certain range
if $mod(t, 2) == 0$ then
    $o_2 = \mathcal{U} + \mathcal{L} - o_1$
else
    $o_2 = sr \times (\mathcal{U}_k + \mathcal{L}_k) - o_1$
end if
$t = t + 1$.
```

A. Test Problems

We use two suites of test problems in our experiments. The first test suite includes nine MTO problems from the CEC 2017 Evolutionary Multi-Task Optimization Competition. Each problem consists of two distinct single-objective optimization tasks, which have their own problem dimensionality (mostly the same except for one problem), global optima, and search ranges. Based on the Spearman's rank correlation coefficient between their respective fitness landscapes, all two tasks in an MTO problem are characterized by high, medium, and low similarities (denoted as HS, MS, and LS) and classified into three categories based on the degree of intersection of their global optima in the unified search space, i.e., complete, partial, and no intersection (denoted as CI, PI, and NI). More details of functions can be referred to [37]. Moreover, test suite 2 contains ten MTO problems, taken from the test suit used in the CEC 2021 Evolutionary Multi-Task Optimization Competition[*] with each problem composed of two distinct single-objective optimization tasks, which bear certain commonality and complementarity in terms of the global optimum and the fitness landscape. These MTO problems possess different degrees of latent synergy between their involved component tasks.

B. Experimental Settings

In this section, we evaluated our proposal and five comparison methods on two benchmarks used in the CEC 2017 and 2021 EMTO competitions. The five EMTO algorithms used in comparison are MFEA [1], MFEA-II [10], MFEA-AKT [38], MFEA-GHS [22] and MTEA-AD [39]. MFEA is the first EMTO method designed specifically for MTO problems, treating each task as a factor influencing population evolution and implicitly ensuring cross-task knowledge transfer via genetic and cultural behavior between offspring and parents. MFEA-II is a multifactorial evolutionary algorithm that uses probabilistic models to effectively adjust the degree of knowledge transfer between tasks. MFEA-AKT is a novel MFEA with adaptive knowledge transfer in which the crossover operator for knowledge transfer across tasks is configured adaptively based on information gathered while the evolutionary search is running online. MFEA-GHS is a novel EMT algorithm that combines MFEA with two complementary strategies: genetic transform and hyper-rectangle search. To mitigate the impact of negative knowledge transfer, the genetic transform strategy is proposed. Mapping vectors are used to convert the individual genetic materials of one task to those of its constitutive task and to create a high-similarity genetic express space among individuals from different tasks. The MTEA-AD algorithm is a new EMT algorithm based on the anomaly detection model. An anomaly detection model for each task is built in each generation to learn the characteristics of the corresponding task. Individuals carrying useful knowledge from other tasks are transferred in this manner. Furthermore, an elitist parameter adaptation strategy is proposed to control the degree of knowledge transfer and effectively filter individuals who carry negative information. All parameter settings are summarized as follows:

a) The population size in all algorithm is 100.

b) The maximum number of function evaluations in our proposal, MFEA, MFEA-II, MFEA-AKT, MFEA-GHS and MTEA-AD is set to $100000 \times K$, where K is the number of tasks.

c) The independent number of runs is configured as 20 for all methods.

d) To maximize the performance of the comparison methods, the probability p_c and the distribution index η_c of SBX are set to 1 and 15, respectively, for MFEA and MFEA-II, to 1 and 2, respectively, for MFEA-AKT and MFEA-GHS. The probability p_m and distribution index η_m of PM are set to $1/D$ and 20, respectively, for MFEA-AKT, to $1/D$ and 5, respectively, for MFEA-GHS, and to $1/D$ and 15, respectively, for MFEA and MFEA-II.

e) The other parameters of these comparison methods are consistent with those of the original papers. For MFEA, the random mating probability (rmp) is set to 0.3. For MFEA-II, the probability model is configured as Normal distribution. For MFEA-AKT, rmp is set to 0.5. For MFEA-GHS and MFEA-DGD, rmp is set to 0.7.

f) For the MFEA-AKT algorithm, parameters in Arithmetical Crossover, Geometrical Crossover, BLX-α Crossover are $\lambda = 2$, $\varpi = 0.25$, $\alpha = 0.03$, respectively.

g) For MFEA-DGD, smoothing parameter $\sigma = 0.01$, $M = 2$, the random matrix $\mathcal{A}$ in each generation is generated in the following form:

$$\mathcal{A} = \frac{1}{2}\begin{pmatrix} 1+\chi & 1-\chi \\ 1-\chi & 1+\chi \end{pmatrix}, \tag{25}$$

where $\chi \sim 0.6 \cdot U(0, 1)$.

h) For parameters of hyper-rectangle search strategy in MFEA-GHS and MFEA-DGD, the range of scaling rate sr is set to be $[0.5, 1.5]$, which implies the value of sr is generated randomly within the range of $[0.5, 1.5]$. Another parameter of MFEA-GHS is the number n_0 of top individuals used to calculate the mapping vectors. The configuration $n_0 = 2$ is used in this experiment.

I) Parameter α in META-AD which controls the frequency of knowledge transfer is set to be 0.1.

[*] http://www.bdsc.site/websites/MTO_competition_2021/MTO_Competition_CEC_2021.html

C. Results and discussion

The comparison results in terms of the mean and standard deviation of the best-achieved FEVs over 20 runs for each component task in each MTO problem from the two test suites are reported in Table II and III, where symbols ”$-$”, ”$\approx$” and ”$+$” imply that the corresponding compared method is significantly worse, similar to, and better than MFEA-DGD on the Wilcoxon rank-sum test with 95% confidence level, respectively. Furthermore, the best performances are indicated in boldface. Next, we analyze the comparison results of MFEA-DGD and other EMTO methods, i.e., MFEA, MFEA-II, MFEA-AKT, MFEA-GHS, and MTEA-AD. MFEA-DGD performs exceptionally well on two benchmarks of continuous MTO problems, as shown in Tables II and III, in terms of the averaged objective value. In comparison to MFEA, our proposal achieves significantly higher solution quality on 16 of 18 tasks in the MTO test suite 1, performs better on 19, ties 1, and loses 0 out of 20 tasks in the MTO test suite 2. Negative transfer is unavoidable in MFEA because knowledge transfer is aimless. However, our proposed algorithm simulates the dynamics of the DGD algorithm, which searches for the gradient descent direction before each knowledge transfer to mitigate the impact of negative transfer. MFEA-DGD obtains better solutions on 15 out of 18 tasks in test suite 1, while MFEA-II outperforms MFEA-DGD on 15, ties 3, and loses 2 out of 20 tasks in test suite 2. This demonstrates that, while MFEA-II optimizes the probability of knowledge transfer between tasks, its ability to overcome negative transfer is still inferior to the algorithm we designed. In terms of the averaged objective value in test suites 1 and 2, our proposal outperforms or matches MFEA-AKT on 16 of 18 tasks and 14 of 20 tasks, respectively. This demonstrates that, while MFEA-AKT can adaptively select the appropriate crossover operator from SBX, Arithmetical, Geometrical, and BLX-$alpha$, the new crossover and mutation operators designed in MFEA-DGD can assist in more effectively searching for the global optimal solution. In comparison to MFEA-GHS, our proposal obtains better solutions on 12 of 18 tasks in test suite 1, while MFEA-DGD performs better on 9, ties 8, and loses 3 out of 20 tasks in test suite 2. MFEA-GHS employs the hyper-rectangle search strategy as well. This comparison result shows that the new operators we proposed outperform the SBX crossover, PM, and genetic transform strategy used in MFEA-GHS. Finally, our proposal outperforms or matches MTEA-AD on 11 of 18 tasks in test suite 1 in terms of averaged objective value, and outperforms MTEA-AD on 12, ties 2, and loses 6 of 20 tasks in test suite 2. MTEA-AD does not outperform MFEA-DGD in problems with extremely high similarity, such as F1, F4, and F7.

Furthermore, the average convergence trends of the five compared methods, namely MFEA-DGD, MFEA, MFEA-II, MFEA-AKT, MFEA-GHS, and MTEA-AD on all ten problems of test suit 2 are shown in Fig. 1 and Fig. 2 to demonstrate the effectiveness of our proposal. Due to space constraints, the average convergence trends of all problems in test suit 1 are shown in the supplementary material in Fig. s1 and Fig. s2. The x-axis represents the number of function evaluations, and the y-axis represents the average objective value on a log scale in these figures. To prevent illegal values on a log scale, we set the averaged objective value of a task to 1E-07 when it is considered solved.

As shown in Fig. 1 and 2, MFEA-DGD has the fastest convergence rate for most tasks. We concentrate on tasks where the average objective value of MFEA-DGD in Table II is comparable to or worse than that of other algorithms. Although the averaged objective value of MFEA-DGD is higher than MTEA-AD, Figure 1 (a) shows that when the number of function evaluations is relatively low, the convergence rate of our proposed algorithm is significantly faster than other algorithms. MTEA-AD, on the other hand, gradually closes the gap as the number of function evaluations increases. For problem 2, while our algorithm performs worse than MTEA-AD statistically with a 95% confidence level, Figure 1 (b) shows that our algorithm's convergence rate is better than MTEA-AD throughout the calculation process, and the speed of fitness descent is extremely fast at the start of the MFEA-DGD. Although MFEA-II performs best for problem 5, Figure 1 (e) shows that the convergence rate of MFEA-DGD is still significantly faster than the remaining five algorithms when the number of function evaluations is not too large. For task 2 in problem 6, by the statistical analysis results, the average objective value of our algorithm is worse than that of MFEA-GHS. However, as shown in Fig. 2 (a), in the initial stage of the algorithm, our algorithm has faster convergence rates than the other five algorithms. For problem 7, the previous statistical analysis shows no significant difference between the average objective value of MFEA-DGD, MFEA-II, MFEA-AKT, MFEA-GHS, and MTEA-AD. However, as shown in Fig. 2 (b), the fitness of MFEA-DGD decreases significantly faster than that of other algorithms. In problem 10, task 2, our algorithm has no significant advantage over MFEA-GHS. However, as shown in Fig. 2 (e), MFEA-DGD has a faster initial convergence rate.

To sum up, the algorithm we propose has excellent efficiency when the number of function evaluations is not very large, which is consistent with theoretical intuition because we know from Theorem 1 that the DGD algorithm has an exponential convergence rate before the optimal individual of each task falls into a ball. As a result, the algorithm is very efficient in utilizing information about the proximity of the optimal solution between similar tasks. The algorithm's convergence rate will slow down only when it is close enough to the optimal solution for each task.

VI. Conclusions

In this paper, we theoretically proved that the DGD method can effectively overcome the non-convex optimization task and has the property of fast convergence. Moreover, we proposed a MFEA-DGD algorithm that extends MFEA by combining new reproduction operators and a hyper-rectangle strategy. The MFEA-DGD is characterized by two novel crossover and mutation operators, which simulate the dynamics of the DGD algorithm, and the OpenAI evolutionary strategy is used to estimate the unknown gradient. The interpretability

TABLE II: The averaged standard objective value of six compared methods, over 20 independent runs on the single-objective MTO test suite 1.

Problem	Task	MFEA-DGD	MFEA	MFEA-II	MFEA-AKT	MFEA-GHS	MTEA-AD
F1:CI+HS	T_1	**1.00E-07±2.65E-23**	2.84E-01±4.23E-02(−)	1.64E-02±6.88E-03(−)	6.38E-02±2.77E-02(−)	5.92E-07±1.22E-06(≈)	9.08E-07±1.01E-07(−)
	T_2	1.44E-06±1.51E-06	5.82E+02±1.06E+2(−)	1.23E+02±2.84E+01(−)	7.61E+01±2.83E+01(−)	8.74E-04±2.09E-03(−)	**9.48E-07±3.46E-08**(≈)
F2:CI+MS	T_1	2.50E-05±1.25E-05	1.04E+01±7.31E+00(−)	1.50E+00±4.54E-01(−)	2.02E+00±5.18E-01(−)	2.14E-03±6.97E-03(−)	**9.56E-07±3.97E-08**(+)
	T_2	**7.40E-07±7.97E-07**	5.65E+02±7.32E+01(−)	1.29E+02±3.03E+01(−)	8.85E+01±3.23E+01(−)	2.32E-02±9.80E-02(−)	9.43E-07±5.58E-08 (−)
F3:CI+LS	T_1	5.02E+00±4.72E+00	3.71E+00±6.40E-01(≈)	**1.38E+00±6.17E-01**(≈)	2.02E+01±9.84E-02(−)	3.44E+00±5.73E-01(≈)	1.34E+01±9.84E+00(−)
	T_2	1.95E+03±3.85E+03	3.80E+03±4.64E+02(−)	2.08E+03±4.49E+02(−)	6.95E+03±9.80E+02(−)	**1.76E+02±1.03E+02**(≈)	5.27E+02±4.72E+02(≈)
F4:PI+HS	T_1	**3.58E-07±4.64E-07**	5.89E+02±1.17E+02(−)	1.54E+02±3.56E+01(−)	3.17E+02±7.09E+01(−)	1.79E+02±1.13E+02(−)	2.73E+02±1.18E+02(−)
	T_2	1.07E-04±3.45E-05	5.28E+00±1.38E+00(−)	2.83E-02±1.21E-02(−)	7.23E-03±7.13E-03(−)	3.54E+00±1.36E+00(−)	**9.25E-07±6.16E-08**(+)
F5:PI+MS	T_1	1.71E+00±5.21E-01	1.82E+01±4.76E+00(−)	1.92E+00±4.34E-01(≈)	1.54E+00±6.08E-01(≈)	2.13E+00±3.01E-01(−)	**9.77E-07±1.61E-08**(+)
	T_2	**1.01E-01±1.23E-01**	6.10E+02±2.44E+02(−)	1.36E+02±2.74E+01(−)	7.64E+01±3.23E+01(−)	4.43E+01±5.97E+01(−)	8.55E+01±5.96E-01 (−)
F6:PI+LS	T_1	7.29E-06±6.42E-06	1.83E+01±4.61E+00(−)	1.91E+00±5.90E-01(−)	2.44E+00±4.65E-01(−)	2.51E-02±4.38E-02(−)	**9.37E-07±4.97E-08**(+)
	T_2	1.90E-03±9.00E-04	1.98E+01±2.23E+00(−)	1.09E+01±2.15E+00(−)	2.54E+00±8.34E-01(−)	1.08E-03±2.97E-03(+)	**9.80E-05±1.02E-04** (+)
F7:NI+HS	T_1	**1.56E+00±2.55E+00**	9.07E+02±8.24E+02(−)	8.86E+02±1.46E+03(−)	1.34E+02±8.06E+01(−)	1.49E+01±1.90E+01(−)	7.62E+01±3.68E+01(−)
	T_2	**5.68E-07±1.13E-06**	5.41E+02±9.90E+01(−)	1.43E+02±3.37E+01(−)	6.78E+01±4.02E+01(−)	3.83E-02±4.37E-02(−)	3.25E+01±7.19E+01(−)
F8:NI+MS	T_1	7.10E-03±1.69E-02	2.79E-01±4.03E-02(−)	1.33E-02±5.10E-03(−)	9.28E-02±3.07E-02(−)	3.21E-03±5.61E-03(≈)	**1.63E-06±6.57E-07**(+)
	T_2	**8.39E-03±4.37E-03**	5.11E+01±4.78E+00(−)	2.37E+01±3.35E+00(−)	1.45E+01±2.25E+00(−)	6.84E-01±4.70E-01(−)	5.35E+00±1.53E+00(−)
F9:NI+LS	T_1	**9.84E-07±2.02E-06**	5.49E+02±1.18E+02(−)	1.38E+02±3.25E+01(−)	4.09E+02±8.11E+01(−)	1.97E+02±2.33E+02(−)	3.12E+02±9.48E+01(−)
	T_2	8.78E+03±1.69E+03	3.67E+03±5.27E+02(+)	2.15E+03±3.19E+02(+)	7.34E+03±9.33E+02(+)	2.42E+03±2.03E+03(+)	**6.69E+02±2.32E+02**(+)
$-/\approx/+$			16/1/1	15/2/1	16/1/1	12/4/2	9/2/7

TABLE III: The averaged standard objective value of six compared methods, over 20 independent runs on the single-objective MTO test suite 2.

Problem	Task	MFEA-DGD	MFEA	MFEA-II	MFEA-AKT	MFEA-GHS	MTEA-AD
1	T_1	6.16E+02±1.56E+00	6.49E+02±3.24E+00(−)	6.33E+02±7.44E+00(−)	6.24E+02±9.55E+00(−)	6.18E+02±3.31E+00(≈)	**6.06E+02±3.88E+00**(+)
	T_2	6.18E+02±1.87E+00	6.48E+02±4.53E+00(−)	6.33E+02±7.58E+00(−)	6.24E+02±9.57E+00(−)	6.18E+02±2.65E+00(≈)	**6.06E+02±3.08E+00**(+)
2	T_1	7.01E+02±1.09E-01	7.06E+02±1.04E+00(−)	7.01E+02±5.82E-02(−)	7.01E+02±1.14E-01(−)	7.01E+02±7.36E-02(−)	**7.00E+02±2.30E-02**(+)
	T_2	7.01E+02±1.63E-01	7.06E+02±1.01E+00(−)	7.01E+02±8.70E-02(−)	7.01E+02±1.08E-01(−)	7.01E+02±6.48E-02(−)	**7.00E+02±1.99E-02**(+)
3	T_1	**6.73E+04±4.00E+04**	5.97E+06±3.24E+06(−)	3.92E+06±2.07E+05(−)	4.37E+06±2.22E+06(−)	2.55E+05±2.10E+05(−)	5.62E+06±2.36E+06(−)
	T_2	**2.53E+05±2.38E+05**	6.89E+06±3.20E+06(−)	3.98E+06±2.24E+06(−)	5.16E+06±2.40E+06(−)	4.43E+05±2.89E+05(≈)	5.24E+06±2.31E+06(−)
4	T_1	**1.30E+03±5.55E-02**	1.30E+03±1.15E-01(−)	1.30E+03±9.68E-02(−)	1.30E+03±1.11E-01(−)	1.30E+03±6.48E-02(−)	1.30E+03±6.17E-02(−)
	T_2	**1.30E+03±7.97E-02**	1.30E+03±5.58E-02(−)	1.30E+03±7.69E-02(−)	1.30E+03±7.76E-02(≈)	1.30E+03±4.84E-02(−)	1.30E+03±5.49E-02(−)
5	T_1	1.57E+03±1.64E+01	1.61E+03±2.45E+01(−)	**1.53E+03±6.93E+00**(+)	1.54E+03±7.16E+00(+)	1.54E+03±8.09E+00(+)	1.53E+03±1.44E+00(+)
	T_2	1.55E+03±1.66E+01	1.62E+03±8.75E+01(−)	**1.53E+03±5.83E+00**(+)	1.54E+03±5.29E+00(+)	1.54E+03±6.27E+00(+)	1.53E+03±1.59E+00(+)
6	T_1	**7.31E+05±2.64E+05**	3.49E+06±1.96E+06(−)	2.77E+06±1.50E+06(−)	1.87E+06±1.15E+06(−)	7.42E+05±6.86E+05(≈)	6.91E+06±6.90E+06(−)
	T_2	7.10E+05±4.23E+05	2.66E+06±1.38E+06(−)	2.32E+06±1.62E+06(−)	2.30E+06±1.31E+06(−)	**4.20E+05±2.67E+05**(+)	1.06E+07±6.91E+06(−)
7	T_1	3.11E+03±3.98E+02	3.34E+03±2.90E+02(−)	**3.08E+03±4.04E+02**(≈)	3.32E+03±4.07E+02(≈)	3.26E+03±3.91E+02(≈)	3.08E+03±4.74E+02(≈)
	T_2	**3.20E+03±3.23E+02**	3.37E+03±4.38E+02(≈)	3.26E+03±3.23E+02(≈)	3.36E+03±3.46E+02(≈)	3.30E+03±3.72E+02(≈)	3.30E+03±4.09E+02(≈)
8	T_1	**5.20E+02±5.06E-02**	5.21E+02±9.12E-02(−)	5.21E+02±4.16E-02(−)	5.21E+02±1.18E-01(−)	5.20E+02±1.11E-01(−)	5.21E+02±3.14E-02(−)
	T_2	**5.20E+02±2.70E-02**	5.21E+02±8.47E-02(−)	5.21E+02±4.46E-02(−)	5.21E+02±1.48E-01(−)	5.20E+02±9.57E-02(−)	5.21E+02±3.46E-02(−)
9	T_1	**8.42E+03±6.68E+02**	8.95E+03±6.86E+02(−)	9.09E+03±2.14E+03(≈)	8.64E+03±9.74E+02(≈)	8.59E+03±7.69E+02(≈)	1.50E+04±2.45E+02(−)
	T_2	**1.62E+03±6.22E-01**	1.62E+03±3.94E-01(−)	1.62E+03±3.48E-01(−)	1.62E+03±4.95E-01(−)	1.62E+03±7.55E-01(−)	1.62E+03±1.72E-01(−)
10	T_1	**1.59E+04±6.06E+04**	6.00E+04±2.56E+04(−)	5.02E+04±2.14E+04(−)	5.00E+04±1.85E+04(−)	2.25E+03±1.07E+04(−)	5.84E+04±1.56E+04(−)
	T_2	1.66E+06±7.79E+05	5.03E+06±2.13E+06(−)	4.50E+06±1.75E+06(−)	3.42E+06±2.00E+06(−)	**1.38E+06±1.20E+06**(≈)	1.77E+07±9.42E+06(−)
$-/\approx/+$			19/1/0	15/3/2	14/4/2	9/8/3	12/2/6

of the role of crossover and mutation operators in MFEA-DGD can be fully derived from DGD's convergence analysis, i.e., the crossover operator combines local convexity between similar tasks, and the mutation operator uses gradient descent to search for better offspring. Furthermore, using the hyper-rectangle strategy broadens the algorithm's search range. On two MTO test suites, we compared MFEA-DGD to some classical or new EMTO algorithms to demonstrate its superiority. Furthermore, we examined the convergence rate scheme to gain an insight into its effectiveness. It should be noted that MFEA-DGD has the same computational complexity as MFEA.

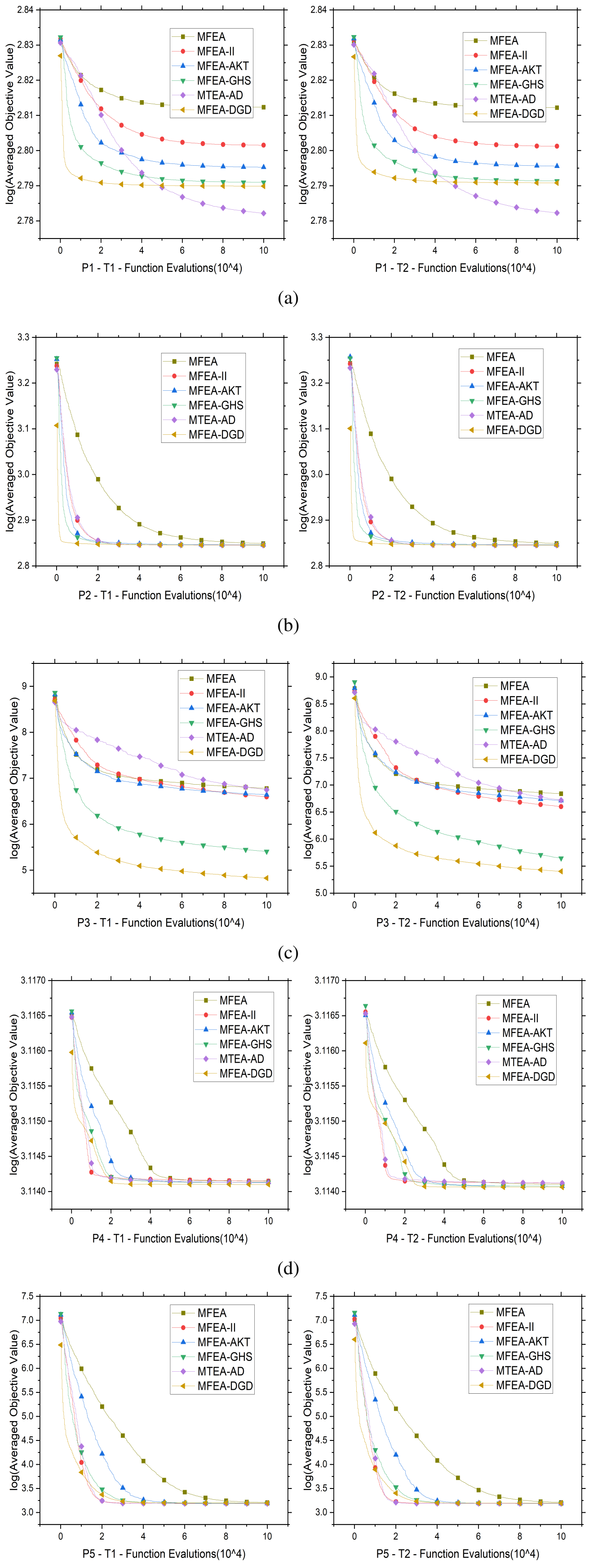
Despite the promising performance of MFEA-DGD, there remains room for further improvement. The performance of MFEA-DGD on MTO problems containing more than two tasks or multi-objective problems should be further investigated. The parallel implementations of MFEA-DGD to speed it up also deserve more effort in future work. The application of MFEA-DGD to real-world problems is also of great potential. The source code of MFEA-DGD written in MATLAB is provided at http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/code.zip.

Appendix: Proof of Theorem 1

A. Distances between $\{\theta_{t+1,i}\}_{i=1}^n$

Lemma 1. *[40, Theorem 3.1] Let* $\mathbf{w} = \{w_i\}$ *be arbitrary vector and* $\boldsymbol{P} = \{p_{ij}\}$ *a stochastic matrix. If* $\mathbf{z} = \boldsymbol{P}\mathbf{w}$, $\mathbf{z} = \{z_i\}$, *then*

$$\max_{h,h'} |z_h - z_{h'}| \leq \tau_1(\boldsymbol{P}) \max_{j,j'} |w_j - w_{j'}|, \tag{26}$$



(a)

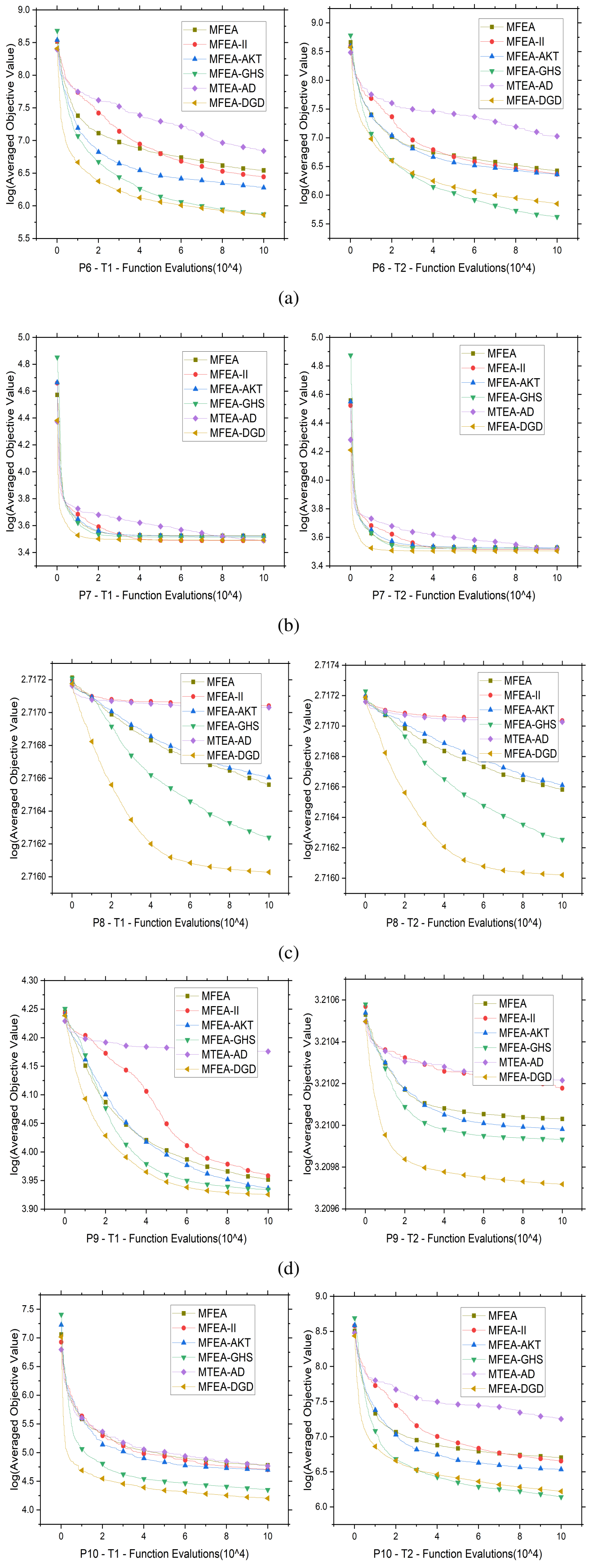
(b)

(c)

(d)

(e)

Fig. 1: Problem 1–5



(a)

(b)

(c)

(d)

(e)

Fig. 2: Problem 6–10

where

$$\tau_1(\boldsymbol{P}) = 1 - \min_{i,j} \sum_k \min\{p_{ik}, p_{jk}\}.$$

Since small distances between $\{\theta_{t+1,i}\}_{i=1}^n$ are desired, in this subsection, we provide the estimation of $\max_{j,j'} \|\theta_{t+1,j} - \theta_{t+1,j'}\|_\infty$ for all $t \geq 0$.

Assume that $\theta'_{t,1}, \ldots, \theta'_{t,n} \in \mathbb{R}^d$ satisfy

$$\mathrm{col}\{\theta'_{t,1}, \ldots, \theta'_{t,n}\} = (\mathcal{A} \otimes I_d)\mathrm{col}\{\theta_{t,1}, \ldots, \theta_{t,n}\},$$

and for $i = 1, \ldots, n$, denote

$$\theta_{t,i} = (\theta_{t,i}[1], \ldots, \theta_{t,i}[d])^\tau, \theta'_{t,i} = (\theta'_{t,i}[1], \ldots, \theta'_{t,i}[d])^\tau.$$

We calculate the distances between $\{\theta_{t+1,i}\}_{i=1}^n$ as follows. For any $k \in [1, d]$, recalling the definition of $\theta'_{t,i}[k]$,

$$\sum_{j=1}^n a_{ij}\theta_{t,j}[k] = \theta'_{t,i}[k], \quad i = 1, \ldots, n. \tag{27}$$

so $(\theta'_{t,1}[k], \ldots, \theta'_{t,n}[k])^\tau = \mathcal{A}(\theta_{t,1}[k], \ldots, \theta_{t,n}[k])^\tau$. Since $\mathcal{A}$ is scrambling doubly stochastic matrix, one has $\tau_1(\mathcal{A}) < 1$. According to Lemma 1,

$$\max_{j,j'} |\theta'_{t,j}[k] - z'_{t,j'}[k]| \leq \tau_1(\mathcal{A}) \max_{j,j'} |\theta_{t,j}[k] - \theta_{t,j'}[k]|,$$

hence

$$\begin{aligned}
\max_{j,j'} \|\theta'_{t,j} - \theta'_{t,j'}\|_\infty &= \max_k \max_{j,j'} |\theta'_{t,j}[k] - \theta'_{t,j'}[k]| \\
&\leq \tau_1(\mathcal{A}) \max_k \max_{j,j'} |\theta_{t,j}[k] - \theta_{t,j'}[k]| \\
&= \tau_1(\mathcal{A}) \max_{j,j'} \|\theta_{t,j} - \theta_{t,j'}\|_\infty. \qquad (28)
\end{aligned}$$

Rewrite (5) as

$$\begin{aligned}
&\mathrm{col}\{\theta_{t+1,1}, \ldots \theta_{t+1,n}\} - \mathrm{col}\{\theta'_{t,1}, \ldots, \theta'_{t,n}\} \\
= \;& -\eta(\mathcal{A} \otimes I_d)\mathrm{col}\{\nabla f_1(\theta_{t,1}), \ldots, \nabla f_n(\theta_{t,n})\},
\end{aligned}$$

we thus deduce

$$\begin{aligned}
&\max_{j,j'} \|\theta_{t+1,j} - \theta_{t+1,j'}\|_\infty \\
\leq \;& \max_{j,j'} \|\theta'_{t,j} - \theta'_{t,j'}\|_\infty + \eta g(\theta_{t,1}, \ldots, \theta_{t,n}) \\
\leq \;& \tau_1(\mathcal{A}) \max_{j,j'} \|\theta_{t,j} - \theta_{t,j'}\|_\infty + \eta g(\theta_{t,1}, \ldots, \theta_{t,n}) \leq \cdots \\
\leq \;& \tau_1^{t+1}(\mathcal{A}) \max_{j,j'} \|\theta_{0,j} - \theta_{0,j'}\|_\infty \\
&+ \eta \sum_{k=0}^t \tau_1^{t-k}(\mathcal{A}) g(\theta_{k,1}, \ldots, \theta_{k,n}) \\
\leq \;& \tau_1^{t+1}(\mathcal{A}) \max_{j,j'} \|\theta_{0,j} - \theta_{0,j'}\|_\infty \\
&+ \frac{\eta}{1 - \tau_1(\mathcal{A})} \max_{k \leq t} g(\theta_{k,1}, \ldots, \theta_{k,n}),
\end{aligned}$$

where $g(\theta_{t,1}, \ldots, \theta_{t,n}) = \max_{j,j'} \|\nabla f_j(\theta_{t,j}) - \nabla f_{j'}(\theta_{t,j'})\|$.

Note that $\theta_{0,1} = \ldots = \theta_{0,n}$, we immediately obtain that for $t \geq 0$,

$$\begin{aligned}
&\max_{j,j'} \|\theta_{t+1,j} - \theta_{t+1,j'}\|_\infty \\
\leq \;& \frac{\eta}{1 - \tau_1(\mathcal{A})} \max_{k \leq t} g(\theta_{k,1}, \ldots, \theta_{k,n}). \qquad (29)
\end{aligned}$$

B. Convergence Analysis

Before the convergence analysis, we provide a technical lemma.

Let $\{A_i; i = 1, \ldots, n\}$ be a sequence of $d \times d$ symmetric matrices satisfying $\|A_i - I_m\| < R$. Denote $I_1(A) \triangleq \mathrm{diag}\{A_1, \ldots, A_n\}$ and $\psi \triangleq (\mathcal{A} \otimes I_d)I_1(A)$.

Lemma 2. *Under Assumption 3, if*

$$\lambda_{\min}\left(\sum_{i=1}^n (I_m - A_i^2)\right) \geq \kappa > 0.$$

there are constants $s \in (0, 1)$ *and* R *determined by* $\mathcal{A}$ *and* ρ *such that*

$$\lambda_{\min}(I_{dn} - \psi^\tau\psi) \geq s\lambda_{\min}\left(\sum_{i=1}^n (I_d - A_i^2)\right).$$

The proof of Lemma 2 is contained in Appendix A.

It is ready to provide the convergence analysis. Define

$$H_i(x) = \int_0^1 \nabla^2 f_i(\theta + \mu(x - \theta))d\mu, \quad x \in \mathbb{R}^d, \tag{30}$$

and

$$\begin{aligned}
Q &= \|\widetilde{\Theta}_0\|^2 + \frac{16\|\nabla f(\theta)\|^2}{s^2\xi^2}, \\
C_0 &= \sup_{\sum_{i=1}^n \|x_i - \theta\|^2 \leq Q} \max_i \|H_i(x_i)\|, \\
C_1 &= \sup_{\sum_{i=1}^n \|x_i - \theta\|^2 \leq Q} g(x_1, \ldots, x_n). \qquad (31)
\end{aligned}$$

Let

$$\eta^* = \min\left\{\frac{1}{4C_0 n}\sqrt{\frac{h}{q}}, \frac{\xi}{100C_0^2 n^{\frac{3}{2}}}\sqrt{\frac{h}{q}}, \frac{\xi\sqrt{1 - \tau_1(\mathcal{A})}}{6\rho C_1}\right\},$$

where h, q are defined in the proof of Lemma 2.

We use induction to show that for all $t \geq 0$,

$$\|\widetilde{\Theta}_t\|^2 \leq Q. \tag{32}$$

It is apparent that (32) holds for $t = 0$. Suppose (32) is true for $t \in [0, k)$, we consider $t = k + 1$. Denote $\mathcal{B} = \mathcal{A}^\tau\mathcal{A}$, by

$$\widetilde{\Theta}_{k+1} = (\mathcal{A} \otimes I_d)(I_{dn} - F_k)\widetilde{\Theta}_k - (\mathcal{A} \otimes I_d)L_k,$$

it yields that

$$\begin{aligned}
&\widetilde{\Theta}_{k+1}^\tau\widetilde{\Theta}_{k+1} \\
= \;& \widetilde{\Theta}_k^\tau(I_{dn} - F_k)(\mathcal{B} \otimes I_d)(I_{dn} - F_k)\widetilde{\Theta}_k \\
&+ 2\widetilde{\Theta}_k^\tau(I_{dn} - F_k)(\mathcal{B} \otimes I_m)L_k + L_k^\tau(\mathcal{B} \otimes I_m)L_k \\
\leq \;& \widetilde{\Theta}_t^\tau(I_{dn} - F_k)(\mathcal{B} \otimes I_d)(I_{dn} - F_k)\widetilde{\Theta}_k \\
&+ 2(1 + \|F_k\|)\sqrt{\widetilde{\Theta}_k^\tau\widetilde{\Theta}_k} \cdot \sqrt{L_k^\tau L_k} + L_k^\tau L_k. \qquad (33)
\end{aligned}$$

By recalling induction assumption we know $\|\widetilde{\Theta}_k\|^2 \leq Q$, so

$$\begin{aligned}
\|F_k\| &\leq \eta \sup_{\sum_{i=1}^n \|x_i - \theta\|^2 \leq Q} \max_i \|H_i(x_i)\| \\
&= \eta C_0 < \frac{1}{2}, \qquad (34)
\end{aligned}$$

which together with (33) leads to

$$\begin{aligned}
& \widetilde{\Theta}_{k+1}^{\tau}\widetilde{\Theta}_{k+1} \\
\le\ & \widetilde{\Theta}_{t}^{\tau}(I_{dn}-F_k)(\mathcal{B}\otimes I_d)(I_{dn}-F_k)\widetilde{\Theta}_k \\
& +3\sqrt{\widetilde{\Theta}_{k}^{\tau}\widetilde{\Theta}_k}\cdot\sqrt{L_k^{\tau}L_k}+L_k^{\tau}L_k. \qquad (35)
\end{aligned}$$

Denote

$$F_k^* = \eta\cdot\mathrm{diag}\{H_{k,1}^*,\dots,H_{k,n}^*\},$$

where $H_{k,i}^* = \int_0^1 \nabla^2 f_i(\theta+\mu(\theta_{k,1}-\theta))d\mu$. Similar to (34), $\|F_k^*\| < \frac{1}{2}$. Furthermore, in view of Assumption 1,

$$\begin{aligned}
\|F_k^* - F_k\| &\le \eta\cdot\max_i \|H_{k,i}^* - H_{k,i}\| \\
&\le \eta\rho\cdot\max_i \|\theta_{k,i}-\theta_{k,1}\| \qquad (36)
\end{aligned}$$

so

$$\begin{aligned}
& \|(I_{dn}-F_k)(\mathcal{B}\otimes I_d)(I_{dn}-F_k) \\
& \quad -(I_{dn}-F_k^*)(\mathcal{B}\otimes I_d)(I_{dn}-F_k^*)\| \\
=\ & \|(F_k^*-F_k)(\mathcal{B}\otimes I_d)(I_{dn}-F_k) \\
& +(I_{dn}-F_k^*)(\mathcal{B}\otimes I_d)(F_k^*-F_k)\| \\
\le\ & \eta\rho\cdot\max_i\|\theta_{k,i}-\theta_{k,1}\|(2+\|F_k\|+\|F_k^*\|) \\
<\ & 3\eta\rho\cdot\max_{i,j}\|\theta_{k,i}-\theta_{k,j}\|. \qquad (37)
\end{aligned}$$

Next, we estimate the first term in the right of (35). To this end, since (29), (37), and $\|\widetilde{\Theta}_l\|^2 \le Q$, $l\in[0,k]$,

$$\begin{aligned}
& \widetilde{\Theta}_{k}^{\tau}(I_{dn}-F_k)(\mathcal{B}\otimes I_d)(I_{dn}-F_k)\widetilde{\Theta}_k \\
\le\ & \widetilde{\Theta}_{k}^{\tau}(I_{dn}-F_k^*)(\mathcal{B}\otimes I_d)(I_{dn}-F_k^*)\widetilde{\Theta}_k \\
& +3\eta\rho\max_{i,j}\|\theta_{t,i}-\theta_{t,j}\|\widetilde{\Theta}_{t}^{\tau}\widetilde{\Theta}_t \\
\le\ & \widetilde{\Theta}_{k}^{\tau}(I_{dn}-F_k^*)(\mathcal{B}\otimes I_d)(I_{dn}-F_k^*)\widetilde{\Theta}_k \\
& +3\rho\frac{\eta^2}{1-\tau_1(\mathcal{A})}\max_{l\le k-1} g(\theta_{l,1},\dots,\theta_{l,n})\widetilde{\|}\widetilde{\Theta}_k\|^2 \\
\le\ & \widetilde{\Theta}_{k}^{\tau}(I_{dn}-F_k^*)(\mathcal{B}\otimes I_d)(I_{dn}-F_k^*)\widetilde{\Theta}_k \\
& +3\rho\frac{\eta^2 C_1}{1-\tau_1(\mathcal{A})}\|\widetilde{\Theta}_k\|^2. \qquad (38)
\end{aligned}$$

Moreover, since Assumption 2 and $\eta n C_0^2 < \frac{\xi}{2}$,

$$\begin{aligned}
& \lambda_{\min}\left(\sum_{i=1}^n [2\eta H_{t,i}^* - \eta^2 (H_{t,i}^*)^2]\right) \\
\ge\ & 2\eta\lambda_{\min}\left(\sum_{i=1}^n H_{t,i}^*\right) - \eta^2 n C_0^2 \\
\ge\ & 2\eta\xi - \eta^2 n C_0^2 > \frac{3\eta\xi}{2}. \qquad (39)
\end{aligned}$$

Let $\kappa = \frac{3\eta\xi}{2}$, it is easy to verify $\eta C_0 < R$, where R is defined as (47) in the proof of Lemma 2. Hence, (34) implies $\|F_k\| < R$, by letting $A_i = I_d - H_{k,i}^*$ in Lemma 2, we obtain

$$\begin{aligned}
& \widetilde{\Theta}_{k}^{\tau}(I_{dn}-F_k^*)(\mathcal{B}\otimes I_d)(I_{dn}-F_k^*)\widetilde{\Theta}_k \\
\le\ & \widetilde{\Theta}_{t}^{\tau}\widetilde{\Theta}_t\left(1-s\eta\lambda_{\min}\left(\sum_{i=1}^n[2H_{t,i}^*-\eta(H_{t,i}^*)^2]\right)\right) \\
\le\ & \widetilde{\Theta}_{t}^{\tau}\widetilde{\Theta}_t\left(1-\frac{3s\eta\xi}{2}\right). \qquad (40)
\end{aligned}$$

Consequently, combine (38) and (40),

$$\begin{aligned}
& \widetilde{\Theta}_{k}^{\tau}(I_{dn}-F_k)(\mathcal{B}\otimes I_d)(I_{dn}-F_k)\widetilde{\Theta}_k \\
\le\ & \widetilde{\Theta}_{t}^{\tau}\widetilde{\Theta}_t(1+3\rho\frac{\eta^2 C_1}{1-\tau_1(\mathcal{A})}-\frac{3\eta\xi}{2}) \\
\le\ & \widetilde{\Theta}_{t}^{\tau}\widetilde{\Theta}_t(1-s\eta\xi), \qquad (41)
\end{aligned}$$

the last inequality attributes to

$$\eta\left(3\rho\frac{C_1}{1-\tau_1(\mathcal{A})}\right) < \frac{\xi}{2}. \qquad (42)$$

Now, by $L_k^{\tau}L_k \le \eta^2\|\nabla f(\theta)\|^2$, (35) and (41),

$$\begin{aligned}
& \widetilde{\Theta}_{k+1}^{\tau}\widetilde{\Theta}_{k+1} \\
\le\ & (1-s\eta\xi)\widetilde{\Theta}_{k}^{\tau}\widetilde{\Theta}_k + 3\eta\|\nabla f(\theta)\|\sqrt{\widetilde{\Theta}_{k}^{\tau}\widetilde{\Theta}_k}+\eta^2\|\nabla f(\theta)\|^2 \\
\le\ & (1-s\eta\xi)Q + 4\eta\|\nabla f(\theta)\|\sqrt{Q}+\eta^2\|\nabla f(\theta)\|^2. \qquad (43)
\end{aligned}$$

By noting that $\eta\|\nabla f(\theta)\| < \sqrt{Q}$ and $s\xi\sqrt{Q} \ge 4\|\nabla f(\theta)\|$,

$$s\xi Q \ge 3\|\nabla f(\theta)\|\sqrt{Q} + \eta\|\nabla f(\theta)\|^2, \qquad (44)$$

then (43) infers $\widetilde{\Theta}_{t+1}^{\tau}\widetilde{\Theta}_{k+1} \le Q$, and hence the induction is completed.

Finally, by (32) and a same analysis as above, we deduce that for $t = 0, 1, \dots$,

$$\begin{aligned}
& \widetilde{\Theta}_{t+1}^{\tau}\widetilde{\Theta}_{t+1} \\
\le\ & (1-s\eta\xi)\widetilde{\Theta}_{t}^{\tau}\widetilde{\Theta}_t + 3\eta\|\nabla f(\theta)\|\|\widetilde{\Theta}_t\| + \eta^2\|\nabla f(\theta)\|^2,
\end{aligned}$$

by taking superior limit in the two side of above inequality, we obtain

$$\begin{aligned}
& \limsup_{t\to+\infty}\|\widetilde{\Theta}_t\|^2 \\
\le\ & (1-s\eta\xi)\limsup_{t\to+\infty}\|\widetilde{\Theta}_t\|^2 \\
& +3\eta\|\nabla f(\theta)\|\limsup_{t\to+\infty}\|\widetilde{\Theta}_t\| + \eta^2\|\nabla f(\theta)\|^2, \qquad (45)
\end{aligned}$$

which implies

$$\limsup_{t\to+\infty}\|\widetilde{\Theta}_t\| \le \frac{\|\nabla f(\theta)\|}{s\xi}(2+\sqrt{3+\eta s\xi}). \qquad (46)$$

Appendix A

Proof of Lemma 2. Denote $\mathcal{B} \triangleq \mathcal{A}^{\tau}\mathcal{A}$. Since $\mathcal{B}$ is irreducible, for any $i\in[1,n-1]$, there is an integer $d_i \ge 2$ and some distinct $c_1^i,\dots,c_{d_i}^i \in [1,n]$ such that

$$\left\{\begin{array}{ll} c_1^i = i, \quad c_{d_i}^i = i+1 & \\ \mathcal{B}[c_j^i, c_{j+1}^i] > 0, \quad j\in[1,d_i-1] \end{array}\right. .$$

Let $q \triangleq \sum_{i=1}^{n-1} d_i - (n-2)$ and define a sequence of $b_j, j = 1,\dots,q$ with $b_1 = c_1^1$ and $b_j = c_{j-\sum_{i=1}^{l}(d_i-1)}^{l+1}$, where $l \in [0, n-2]$ and

$$1+\sum_{i=1}^{l}(d_i-1) < j \le 1+\sum_{i=1}^{l+1}(d_i-1).$$

Hence $\mathcal{B}[b_j, b_{j+1}] > 0$ for all $j\in[1,q-1]$.

Denote $h = \min_{j\in[1,q-1]} \mathcal{B}[b_j, b_{j+1}]$. Select

$$R^2 = \min\left\{\frac{h}{16n^2q}, \frac{\kappa}{144n}\sqrt{\frac{h}{nq}}\right\}, \tag{47}$$

$$s = \min\left\{\frac{nR^2}{\kappa}, \frac{1}{8n}\right\}, \tag{48}$$

Now, suppose for a constant vector $x \in \mathbb{R}^{mn}$ with $\|x\| = 1$,

$$x^\tau(I_{dn} - \psi_1^\tau \psi_1)x < s\kappa. \tag{49}$$

Write $x = \mathrm{col}\{x_1, \ldots, x_n\} \in \mathbb{R}^{dn}$. A direct calculation yields

$$\begin{aligned} x^\tau(I_{dn} - \psi^\tau\psi)x \;&=\; \sum_{1\le i<j\le n} \mathcal{B}[i,j](\|A_i z_i - A_j z_j\|^2) \\ &\quad +x^\tau(I_{dn} - I_1^2(A))x, \end{aligned} \tag{50}$$

Note that

$$\begin{aligned} &\sum_{j=1}^{q-1} \mathcal{B}[b_j, b_{j+1}]\|x_{b_j} - x_{b_{j+1}}\|^2 \\ \le\;& \sum_{1\le i<j\le n} \mathcal{B}[i,j]\|x_i - x_j\|^2 \\ \le\;& 2\sum_{1\le i<j\le n} \mathcal{B}[i,j](\|A_i x_i - A_j x_j\|^2 \\ & +\|(I_d - A_i)x_i - (I_d - A_j)x_j\|^2) \\ \le\;& 2s\kappa + 2nR^2. \end{aligned} \tag{51}$$

By *Cauchy-Schwarz inequality* and (51),

$$\begin{aligned} \|x_i - x_j\|^2 \;&\le\; q\sum_{j=1}^{q-1}\|x_{b_j} - x_{b_{j+1}}\|^2 \\ &<\; \frac{2q(s\kappa + nR^2)}{\min_{j\in[1,q-1]}\mathcal{B}[b_j, b_{j+1}]}, \quad i<j. \end{aligned} \tag{52}$$

The definition of R, s, and (52) yield that $\|x_1 - x_i\|^2 \le \frac{1}{4n}$ for all $i > 1$. Since $\sum_{i=1}^n \|x_i\|^2 = 1$,

$$\|x_1\|^2 \ge \frac{1}{2n-1} - \frac{1}{4n} > \frac{1}{4n}.$$

Therefore,

$$\begin{aligned} & x^\tau(I_{dn} - \psi^\tau\psi)x \ge x^\tau(I_{dn} - I_1^2(A))x \\ =\;& \sum_{i=1}^n x_i^\tau(I_d - A_i^2)x_i \\ \ge\;& x_1^\tau\sum_{i=1}^n(I_d - A_i^2)x_1 - \sum_{i=2}^n \|I_m - A_i^2\|\|x_1 - x_i\|^2 \\ & -2\sum_{i=2}^n\|I_d - A_i^2\|\|x_1 - x_i\| \\ \ge\;& \frac{\kappa}{4n} - 9R\sqrt{\frac{2q(s\kappa + nR^2)}{h}} \ge \frac{\kappa}{4n} - 9R\sqrt{\frac{4qnR^2}{h}} \\ \ge\;& \frac{1}{8n}\kappa \ge s\kappa, \end{aligned}$$

which contradicts to (49). So,

$$x^\tau(I_{dn} - \psi^\tau\psi)x \ge s\kappa$$

holds for all unit vector $x \in \mathbb{R}^{dn}$ and Lemma 2 follows. ■

References

[1] A. Gupta, Y.-S. Ong, and L. Feng. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2016.

[2] Y.-S. Ong and A. Gupta. Evolutionary multitasking: A computer science view of cognitive multitasking. *Cognitive Computation*, 8(2):125–142, 2016.

[3] R. Chandra, A. Gupta, Y.-S. Ong, and C.-K. Goh. Evolutionary multitasking: A computer science view of cognitive multitasking. *Neural Processing Letters*, 47(3):993–1009, 2018.

[4] L. Feng, Y.-S. Ong, S. Jiang, and A. Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, 21(5):760–772, 2017.

[5] B. Da, A. Gupta, Y. S. Ong, and L. Feng. The boon of gene-culture interaction for effective evolutionary multitasking. *Australasian Conference on Artificial Life and Computational Intelligence*, page 54–65, 2016.

[6] A. Gupta and Y.-S. Ong. Genetic transfer or population diversification? deciphering the secret ingredients of evolutionary multitask optimization. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, page 1–7, 2016.

[7] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha. Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, page 1–8, 2016.

[8] R. Chandra, Y.-S. Ong, and C.-K. Goh. Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction. *Neurocomputing*, 243:21–34, 2017.

[9] P. D. Thanh, H. T. T. Binh, and T. B. Trung. An efficient strategy for using multifactorial optimization to solve the clustered shortest path tree problem. *Applied Intelligence*, 50(4):1–26, 2020.

[10] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II. *IEEE Transactions on Evolutionary Computation*, 24(1):69–83, 2019.

[11] K. K. Bali, A. Gupta, Y.-S. Ong, and P. S. Tan. Cognizant multitasking in multiobjective multifactorial evolution: MO-MFEA-II. *IEEE Transactions on Cybernetics*, 51(4):1784–1796, 2020.

[12] Y.-W. Wen and C.-K. Ting. Parting ways and reallocating resources in evolutionary multitasking. *2017 IEEE Congress on Evolutionary Computation (CEC)*, page 2404–2411, 2017.

[13] K. K. Bali, A. Gupta, L. Feng, Y.-S. Ong, and T. P. Siew. Linearized domain adaptation in evolutionary multitasking. *2017 IEEE Congress on Evolutionary Computation (CEC)*, page 1295–1302, 2017.

[14] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y.-S. Ong, K.-C. Tan, and A. Qin. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 49(9):3457–3470, 2018.

[15] G. Li, Q. Lin, and W. Gao. Multifactorial optimization via explicit multipopulation evolutionary framework. *Information Sciences*, 512:1555–1570, 2020.

[16] J. Ding, C. Yang, Y. Jin, and Y. Chai. Generalized multi-tasking for evolutionary optimization of expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):44–58, 2019.

[17] H. Han, X. Bai, Y. Hou, and J. Qiao. Self-adjusting multi-task particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 26(1):145–158, 2021.

[18] L. Bai, W. Lin, A. Gupta, and Y.-S. Ong. From multitask gradient descent to gradient-free evolutionary multitasking: A proof of faster convergence. *IEEE Transactions on Cybernetics*, page 145–158, 2021.

[19] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.

[20] K. Deb and M. Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26(4):30–45, 1996.

[21] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[22] Z. Liang, J. Zhang, L. Feng, and Z. Zhu. A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking. *Expert Systems with Applications*, 138(30), 2019.

[23] F. S. Cattivelli and A. H. Sayed. Diffusion lms strategies for distributed estimation. *IEEE Transactions on Signal Processing*, 58(3):1035–1048, 2009.

[24] J. Chen, C. Richard, and A. H. Sayed. Diffusion lms over multitask networks. *IEEE Transactions on Signal Processing*, 63(11):2733–2748, 2015.

[25] S. Vlaski and A. H. Sayed. Distributed learning in non-convex environments—part I: Agreement at a linear rate. *IEEE Transactions on Signal Processing*, 69:1242–1256, 2021.

[26] S. Vlaski and A. H. Sayed. Distributed learning in non-convex environments—part II: Polynomial escape from saddle-points. *IEEE Transactions on Signal Processing*, 69:1257–1270, 2021.

[27] Y. Nesterov and V. Spokoiny. Random gradient-free minimization of convex functions. *IEEE Transactions on Signal Processing*, 17(2):527–566, 2017.

[28] K. Choromanski, M. Rowland, V. Sindhwani, and R. E. Turner. Structured evolution with compact architectures for scalable policy optimization. *International Conference on Machine Learning*, page 969–977, 2018.

[29] H. R. Tizhoosh. Opposition-based learning: A new scheme for machine intelligence. *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, 1:695–701, 2005.

[30] M. El-Abd. Opposition-based artificial bee colony algorithm. *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, page 109– 116, 2011.

[31] Y. Zhou, J. Hao, and B. Duval. Opposition-based memetic search for the maximum diversity problem. *IEEE Transactions on Evolutionary Computation*, 21(5):731–745, 2017.

[32] X. Ma, F. Liu, Y. Qi, M. Gong, M. Yin, L. Lin, L. Jiao, and J. Wu. MOEA/D with opposition-based learning for multiobjective optimization problem. *Neurocomputing*, 146:48–64, 2014.

[33] S. Xie and L. Guo. A necessary and sufficient condition for stability of lms-based consensus adaptive filters. *Automatica*, 93:12–19, 2018.

[34] G. Pavai and T. V. Geetha. A survey on crossover operators. *ACM Computing Surveys (CSUR)*, 49(4):72, 2016.

[35] J. C. Bongard. A probabilistic functional crossover operator for genetic programming. *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, page 925–932, 2010.

[36] X. Qiu, K. C. Tan, and J.-X. Xu. Multiple exponential recombination for differential evolution. *IEEE Transactions on Cybernetics*, 47(4):995–1006, 2017.

[37] B. Da, Y. S. Ong, L. Feng, A. K. Qin, A. Gupta, Z. Zhu, C. K. Ting, K. Tang, and X. Yao. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metrics and baseline results. *arXiv preprint arXiv:1706.03470*, 2017.

[38] L. Zhou, L. Feng, K. C. Tan, J. Zhong, Z. Zhu, K. Liu, and C. Chen. Toward adaptive knowledge transfer in multifactorial evolutionary computation. *IEEE Transactions on Cybernetics*, 51(5):2563–2576, 2020.

[39] C. Wang, J. Liu, K. Wu, et al. Solving multi-task optimization problems with adaptive knowledge transfer via anomaly detection. *IEEE Transactions on Evolutionary Computation*, 2021.

[40] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer Series in Statistics, 1981.