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. Despite the effectiveness of MFEA in solving difficult optimization problems, there is no evidence of population convergence or theoretical explanations of how knowledge transfer increases algorithm performance. To fill this gap, we propose a new MFEA based on diffusion gradient descent (DGD) namely MFEA-DGD in this paper. We prove the convergence of DGD for multiple similar tasks and show that the local convexity of some tasks can help other tasks escape from local optimum by knowledge transfer. On this theoretical foundation, we design new complementary crossover and mutation operators in MFEA-DGD, such that the evolution population is endowed with a dynamic equation similar to DGD, i.e., the convergence is guaranteed and the benefit from knowledge transfer is explainable. A hyper-rectangular search strategy is also introduced to allow MFEA-DGD to explore more underdeveloped areas in the unified express space of all tasks and the subspace of each task. MFEA-DGD is verified on various multi-task optimization problems and the experimental results demonstrate 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.**

I. INTRODUCTION

Evolutionary multitasking (EMT) [1], [2] solves 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]. Taking advantage of knowledge transfer, EMT has been shown to outperform single-task EAs on various optimization problems [5]–[9] and real-world applications [10]–[14].

Multifactorial evolutionary algorithm (MFEA) [1] represents the first attempt of EMT and has received fast increasing attention thanks to its simplicity and efficiency. Many MFEA improvements or variants have been surveyed in [15], [16]. The main ideas of some representative algorithms [17]–[31] are summarized in Table I for the convenience of the reader. MFEA was also extended to solve many-task problems, where more than two optimization tasks are considered [32]–[40] , and multi-objective multi-task problems with each task being a multi-objective optimization problem [41]–[48].

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.* [20] presented a pioneer attempt to prove the algorithm convergence and analyze the effects of inter-task interactions in MFEA using probability distribution to model the population. The convergence of the used probability distribution requires the updating of probability density at each point in the search space, which strictly implies the entire space must be searched with a population of infinite size. This underlying assumption might be unrealistic in MFEAs. For other EMT algorithms, Bali *et al.* [49] also presented a convergence-guaranteed multi-task gradient descent algorithm. The convergence proof works well for convex problems, whereas it might not hold for non-convex tasks that are more common in practical applications. Han *et al.* [50] presented a convergence analysis of a particle swarm optimization based EMT algorithm, which suffers the same issue as [20].

In this study, we propose a new MFEA based on diffusion gradient descent (DGD), namely MFEA-DGD, which enables a more general theoretical explanation of knowledge transfer and population convergence. We firstly describe the motivation of using DGD and then theoretically prove the 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) [51] and polynomial mutation (PM) [52] 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 [53], [54] to simulate the gradient. Moreover, the hyper-rectangular search strategy inspired by [21] is introduced to MFEA-DGD to enable the exploration of more undeveloped areas. MFEA-DGD is compared with other state-of-the-art MFEAs and EMT algorithms on multi-task optimization problems and

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)

TABLE I: Summary of some representative EMT algorithms

Algorithm	Main Ideas	Reference
ASCMFDE	Intertask knowledge transfer in the low-dimension subspaces	[26]
LDA-MFEA	Linearized domain adaptation strategy	[17]
MFEARR	Detection of parting ways and reallocating fitness evaluations	[18]
G-MFEA	Decision variable translation and shuffling	[19]
MFEA-II	Online random mating parameter learning	[20]
MFEA-GHS	Genetic transform and hyper-rectangle search	[21]
SREMTO	Self-Regulated knowledge transfer	[22]
MTO-DRA	Online dynamic resource allocation	[23]
MFMP	Genetic transfer based on multi-population	[24]
MFEA-AKT	Adaptive knowldege transfer	[25]
MTEA-AD	Anomaly detection model	[27]
GFMFDE	Knowledge transfer based on active coordinate system	[28]
MKTDE	Meta knowledge transfer	[29]
SA-MTPSO	Knowledge estimation metric and self-adjusting knowledge transfer	[30]
MFEA/MVD	Multi-objective decomposition based helper-task	[31]

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 demonstrated theoretically for the first time. Under some fundamental assumptions, we demonstrate that DGD is effective for non-convex problems and can rapidly converge near global optimum via 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 the performance of algorithms for similar tasks. This concept is easily adaptable to other types of decentralized optimization algorithms and EMT algorithm design.
- We estimate the convexity of each pair of twin tasks in the benchmark and calculate the distance between their global optimums, further verifying the consistency of experimental and theoretical results.

The rest of the paper is organized as follows. Section II introduces some preliminaries on MFEA and DGD. 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 II below provides a summary of the symbols used in this article.

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, and then demonstrate the principle of the DGD algorithm, which together with OpenAI ES derives the crossover and mutation operators in MFEA-DGD.

TABLE II: Notation conventions used in this article

$\mathbf{1}_k$	k-dimensional vector with all elements being 1.
$\mathbb{R}^{k\times k}$	$k \times k$ real matrices.
A^T	Transpose of A.
$U(a,b)$	Uniform distribution on (a,b).
$\mathrm{col}\{a,b\}$	Column vector with entries a and b.
$\mathrm{diag}\{a,b\}$	Diagonal matrix with entries a and b.
$\|x\|$	Euclidean norm of its vector argument.
$\|A\|$	2-induced norm of matrix A (its largest singular value).
I_k	Identity matrix of size $k \times k$.
$\mathcal{L}$	Lebesgue measure.
$f : X \mapsto Y$	f is a function with domain X and codomain Y.
$\mathbb{N}^+$	Positive integers.
$a = O(b)$	There is a constant $C > 0$ such that $\lvert a\rvert \le C \cdot b$.

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 express 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 $\varphi_p = 1/r_p^{\tau_p}$.

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

Algorithm 1 The Framework of MFEA

Input: N (population size), n (number of tasks)
Output: a series of solutions

1: Initialize the population P
2: Randomly assign the skill factor for each individual in P
3: Evaluate factorial cost of each individual
4: **while** the stopping criteria are not reached **do**
5: Generate offspring population O based on assortative mating
6: Perform vertical cultural transmission
7: Evaluate offspring individuals
8: Generate new population $P' = P \cup O$
9: Update the scalar fitness φ and skill factor τ of each individual
10: Select the N fittest individuals from P' to form P
11: **end while**

Algorithm 2 Diffusion Gradient Descent (DGD)

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

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.

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 optimum. To escape local optimums by using useful information between similar tasks, we consider diffusion strategies [55] 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 [56].

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 [56], the general structure of DGD algorithm consists of the following steps:

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

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.,

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

The appropriate selection of $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{C}$ leads to several adaptive strategies as special cases of (3). With the setting $\mathcal{A}_1 = I_n$, we get the so-called adapt-then-combine (ATC) DGD. Moreover, 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 standard gradient descent without any knowledge transfer. According to [55], the ATC diffusion LMS algorithm tends to outperform the CTA version. Notice that LMS is essentially a gradient descent of a quadratic optimization problem, so we adopt the ATC version of DGD in this paper. To facilitate the follow-up study, we consider a common case [57] 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 $\mathcal{A}_1 = \mathcal{A}_2 = \{a_{ij}\}_{n \times n}$.

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

In this section, we first introduce a new theoretical result 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 ES.

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. Moreover, 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 $M = \{m_{ij}\} \in \mathbb{R}^{n\times n}$ is said to be irreducible, if there is a sequence $i_1, \ldots, i_l$ contains $\{1, \ldots, n\}$, 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:*

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

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

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

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

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

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

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

Remark 1. *Assumption 1 is a conventional condition in non-convex optimization theory. Assumption 2 represents joint strong convexity in some sense, allowing for multiple non-convex tasks among n tasks. Assumption 2 will be easily satisfied as long as some of the tasks are well convex. Assumption 3 means the connectivity of information exchange between tasks, and we can easily set parameters to make this condition hold.*

Define $\widehat{\Theta}_t = \text{col}\{\theta_{t,1} - \theta_1^*, \ldots, \theta_{t,n} - \theta_n^*\}$, where

$$\theta_i^* = \underset{\theta}{\text{argmin}}\, f_i(\theta), \quad i = 1, \ldots, n.$$

We have following result.

Theorem 1. *Under Assumption 1–3, there is $\eta^* > 0$ such that for any $\eta \in (0, \eta^*)$ and initial value $\theta_{0,1} = \ldots = \theta_{0,n}$,*

$$\begin{aligned} &\sum_{i=1}^n \left(f_i(\theta_{t,i}) - f_i(\theta_i^*)\right) \le \ell \|\widehat{\Theta}_t\|^2 \\ \le\;& 2\left(\frac{50^2 n^2 \ell^2}{\xi^2} + 1\right) \ell \sum_{i=2}^n \|\theta_1^* - \theta_i^*\|^2 \\ &+2\left(1 - \frac{\eta\xi}{16n}\right)^t \ell \sum_{i=1}^n \|\theta_{0,i} - \theta_1^*\|^2. \end{aligned}$$

The proof of this theorem is provided at http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/proof.pdf. It is worth mentioning here that if θ is restricted to a bounded region, the conclusion of Theorem 1 still holds and the proof will be simpler because one of the difficulties in the unbounded region setting lies in proving that the estimation error $\|\widehat{\Theta}_t\|$ falls into a bounded region in advance.

Remark 2. *(Convergence) Theorem 1 implies that there is $\beta \in (0, 1)$ such that $\|\widehat{\Theta}_t\| = O(\epsilon) + O(\beta^t)$ when $\max_i \|\theta_1^* - \theta_i^*\| < \epsilon$. In words, under evolutionary strategy (5), the similarity of global optimums among n tasks ($\max_i \|\theta_1^* - \theta_i^*\|$ is small) can deduce that the DGD algorithm converges geometrically to a neighborhood near the global optimums.*

Remark 3. *(Interpretability) Observing Assumption 2, we do not need the convexity of each task. For example, even if $f_1, \ldots, f_{n-1}$ are non-convex functions, as long as the convexity of f_n is enough to ensure the hessian matrix of f^{glob} is positive definite, the algorithm can still converge near the optimums. Therefore, as long as there is one task with good convexity in a group of similar tasks, it can provide more useful information about gradient to help other tasks approach their respective optimums. This indicates that the DGD algorithm provides a strong interpretability for knowledge transfer.*

Remark 4. *It is worth mentioning that matrix $\mathcal{A}$ in Theorem 1 can be replaced by the time-varying $N \times N$ matrix $\mathcal{A}_t = \{a_{t,li}\}$, the gradient of functions $\nabla f_1, \ldots, \nabla f_n$ can also be replaced the time-varying function group $\nabla f_{t,1}, \ldots, \nabla f_{t,N}$, where each $f_{t,i} \in \{f_1, \ldots, f_n\}$. The theoretical interpretation of the evolutionary algorithm we designed in later sections hinges heavily on this case. In this case, a similar conclusion to Theorem 1 can still be established, assuming there is integer $m > 0$ such that $\sum_{j=t}^{t+m} \sum_{i=1}^N \nabla^2 f_{j,i} \ge \xi I_d$, each $B_t = \mathcal{A}_{t+m}\mathcal{A}_{t+m-1}\cdots\mathcal{A}_t$ satisfies Assumption 3 and the non-zero entries of the matrices $\mathcal{A}_t$ have the uniform lower bound a:*

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

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 [58], which is very similar to our assumptions here.

B. New crossover and mutation operators

Since DGD has fast 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 EAs. In the last few decades, a large number of crossover operators have been proposed in the literature for a wide range of optimization problems [59]–[61]. 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:

$$\begin{cases} c_1^i = \nu \cdot p_1^i + (1-\nu) \cdot p_2^i \\ c_2^i = (1-\nu) \cdot p_1^i + \nu \cdot p_2^i \end{cases}. \tag{6}$$

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:

$$\mathbf{c} = \mathcal{A} \cdot \mathbf{p}. \tag{7}$$

If the population size is set to $N = 2$, 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

$$\mathbf{p}_t = \mathcal{A}_{t-1} \cdot \mathcal{A}_{t-2} \cdots \mathcal{A}_0 \cdot \mathbf{p}_0. \tag{8}$$

Such a formula cannot guarantee the convergence rate of the population to the global optimums, 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 (7):

$$\mathbf{c} = \mathcal{A} \cdot \mathbf{p}' \approx \mathcal{A}(\mathbf{p} - \eta \nabla \mathbf{f}(\mathbf{p})), \tag{9}$$

where $\mathbf{f}$ is a two-dimensional vector valued function, of which each component belongs to the set $\{f_1, \ldots, f_n\}$, and the gradient $\nabla \mathbf{f}$ can be approximated by enhanced OpenAI evolutionary strategy referred in [54].

In the literature, PM is one of the most frequently used mutation operators for producing offspring by randomly mutating parents. Because natural evolution is influenced by environmental factors and is not entirely random, we anticipate that the mutation will have a directionality. Parental mutations require empirical information from other individuals. Theoretically, the mutation operator was designed using GD. We expect the offspring $\mathbf{c}$ to be produced in a quasi-gradient descent direction,

$$\mathbf{c} \approx \mathbf{p} - \eta \nabla \mathbf{f}(\mathbf{p}), \tag{10}$$

which is a special case of (9) with $\mathcal{A} = I_2$.

We apply the crossover (9) and mutation (10) 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

$$P_t \approx \mathcal{A}_{t-1}(P_{t-1} - \eta \nabla \mathbf{f}_{t-1}(P_{t-1})), \tag{11}$$

where

$$\begin{aligned} \mathcal{A}_{t-1} &= \operatorname{diag}\{A_{t-1,1}, \ldots, A_{t-1,N_1/2}, \underbrace{I_2, \ldots, I_2}_{(N-N_1)/2}\}, \\ \mathbf{f}_{t-1} &= (f_{t-1,1}, \ldots, f_{t-1,N})^T, \end{aligned} \tag{12}$$

and $A_{t-1,i} \in \mathbb{R}^{2\times 2}$, $i = 1, \ldots, N/2$, $f_{t-1,i} \in \{f_1, \ldots, f_n\}$, $i = 1, \ldots, N$. Equation (11) 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 (9) and (10), respectively, the advantage of DGD can be retained by choosing proper $\mathcal{A}_i$, $i \geq 0$ (as described in Remark 4). 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.

Algorithm 3 MFEA-DGD

Input: N (population size), n (number of tasks), M(number of individuals to simulate the gradient), σ(smoothing parameter)

Output: a series of solutions

Initialize population P; Randomly assign skill factor τ 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 $\eta = 1/L$
    Randomly select two parent individuals p_1, p_2
    Randomly generate a matrix $\mathcal{A} \in \mathbb{R}^{2\times 2}$
    Obtain skill factor τ_1, τ_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
  Generate new population $NP = P \cup O$
  Select fittest individuals from NP to form P
  Update learning late η
end while

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, where the functions $Quasi - GradMutation()$, $GradTransform()$ and $Hyper-rectangleSearch()$ are defined in Algorithms 4-6, respectively. Generally, the main difference between MFEA-DGD and the conventional MFEA 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 have the same skill factor, the gradient transform and hyper-rectangle search strategies produce o_1 and o_2, respectively. When p_1 and p_2 have different skill factors, there is still a random mating probability (rmp) for the two strategies to be activated. Otherwise, they generate offspring individuals o_1 and o_2 via quasi-gradient descent mutation operator, respectively. Here, rmp is used to adjust the frequency of information exchange between tasks. When 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 the optimization process of GD 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 updated learning rate (or step size) η can be used to enhance the performance of the designed operator.

It is worth mentioning that the hyper-rectangle search strategy is not our original method (see Algorithm 6). The purpose of adopting this strategy is to expand the search area of the population and increase the diversity of the population, so this part is independent from the new operators. There may actually exist better strategies to enhance the diversity of the population, so the proposed algorithm has a great potential for improvement.

In one generation, MFEA-DGD involves three main steps, gradient transform, quasi-gradient descent mutation, and hyper-rectangle search. Each step takes $O(Nd)$ time, where N and d refer to the total size of the population, and the max number of decision variables in all tasks, respectively. An elitism-based parameter adaptation strategy has a time complexity of $O(N\log(N/n))$ (for example, fast sorting), where n is the number of tasks. To summarize, the computational complexity of MFEA-DGD in one generation is $O(N(d+\log(N/n)))$.

The following subsections describe in detail the critical components of MFEA-DGD, including the gradient transform, quasi-gradient descent mutation, and the update criteria for learning rate.

B. Quasi-gradient descent mutation

In contrast to conventional mutation, we design mutation criteria by approximating gradient descent with a number of function evaluations. Consistent with our intuitive understanding of evolutionary processes, this can be interpreted as individuals choosing the appropriate evolutionary direction based on the experiences of randomly selected individuals around them. The pseudo code is given in Algorithm 4.

In Algorithm 4, $\{\epsilon_j\}_{j=1}^M$ are generated marginally by standard Gaussian distribution $N(0, I_d)$. $\nabla f_\sigma(p)$ indicates the level of mutation at p. When there is a large difference in fitness between two individuals $p+\sigma\xi_j$ and $p-\sigma\xi_j$, it is demonstrated that the direction of mutation is likely to decrease p's fitness more quickly, with p gaining more empirical information from randomly generated individuals $p+\sigma\xi_j$ and $p-\sigma\xi_j$. In contrast, if the fitness gap between two individuals is small, the level of mutation will be relatively low.

Algorithm 4 The Quasi-gradient Mutation Strategy

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

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 σ, learning rate η, matrix $\mathcal{A} \in \mathbb{R}^{2\times 2}$
Output: the generated child p_3
1: **for** $i = 1, 2$ **do**
2: $\quad\nabla f_{\sigma,i}(p_i) = \sum_{j=1}^M \frac{\xi_i^j f_{\tau_i}(p_i+\sigma\xi_i^j)-\xi_i^j f_{\tau_i}(p_i-\sigma\xi_i^j)}{\sigma}$
3: **end for**
4: $p_3 = a_{11}(p_1-\eta\nabla f_{\sigma,1}(p_1)) + a_{12}(p_2-\eta\nabla f_{\sigma,2}(p_2))$

C. Gradient transform strategy

The majority of offspring during evolution are produced by parents who excel at different tasks. If the parents are skilled at tasks that are significantly unrelated, the offspring may not well-adapted to either task. Consequently, they have a difficult time surviving to the next generation, and the efficiency of knowledge transfer between tasks decreases. The Algorithm 5 provides pseudo code for the proposed gradient transform strategy to improve knowledge transfer efficiency.

Next, we take an example of two-task problem as shown in Fig. 1 to illustrate the motivation of the proposed algorithm. Given two parent individuals p_1 and p_2, two offspring $\tilde{p}_1$ and $\tilde{p}_2$ are produced by applying quasi-gradient descent to them, respectively. The third offspring p_3 is produced by combining these two offspring linearly (crossover). As can be seen from Fig. 1 that Ackley function is non-convex and $\tilde{p}_1$, which we obtained after using quasi-gradient descent for p_1, lies at the local optimum of Ackley. Although $\tilde{p}_1$ has a better fitness than p_1, $\tilde{p}_1$ is far from the global optimum of Ackley. On the other hand, the Sphere function is convex, so using quasi-gradient descent on p_2 (i.e., to produce $\tilde{p}_2$) can effectively approximate the global optimum of Sphere function. Note that the global optimum of Ackley and Sphere are in similar locations, so a linear combination of $\tilde{p}_1$ and $\tilde{p}_2$ can help $\tilde{p}_1$ escape from the local optimum and search toward the global optimum of Ackley to reach the location of p_3.

Essentially, the strategy suggests using a gradient descent direction that converges to the global optimum to counteract the effects of slow gradient descent or misdirection. By

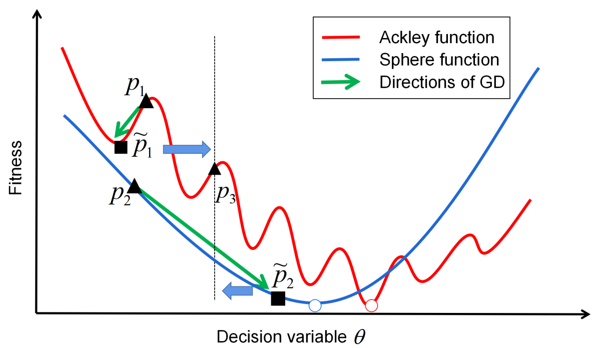

Fig. 1: Illustration of gradient transform strategy, where triangles and squares represent the parents and the solution obtained by the parent using gradient descent, respectively. The curves of Ackley and Sphere are distinguished with red and blue, respectively. The global optimums are supposed to be located in the circles.

exchanging gradient information between different tasks, the strategy can move away from the local optimums and converge to the global optimums.

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 learning rate. For each direction ξ_i^j, the values $\{f_{\mathcal{T}_i}(p_i+\sigma\xi_i^j), f_{\mathcal{T}_i}(p_i-\sigma\xi_i^j)\}$, are used to estimate the directional local Lipschitz constants of $\nabla f_{\mathcal{T}_i}$ by

$$L_i^j = \left| \frac{f_{\mathcal{T}_i}(p_i+\sigma\xi_i^j) + f_{\mathcal{T}_i}(p_i-\sigma\xi_i^j) - 2f_{\mathcal{T}_i}(p_i)}{\sigma^2} \right| .$$

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

$$L \longleftarrow (1-\gamma)L_D + \gamma L, \quad \eta = \sigma/L, \tag{13}$$

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 primarily account for the fact that the best learning rate for gradient descent is typically equal to the Lipschitz constant of the function's gradient, see page 29 of [62].

V. Experiments

In previous sections, we have claimed that MFEA-DGD inherits the advantages of DGD, and is highly interpretable to effectively solve multi-task non-convex optimization problems. To verify these conclusions, in this section we design experiments and discuss the results to answer several questions as follows.

1) Q1: Does MFEA-DGD perform competitively compared to existing state-of-the-art EMT algorithms?
2) Q2: Can the performance of MFEA-DGD be explained by theory?

Algorithm 6 The Hyper-rectangle Search Strategy [21]

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
1: Generate a random number sr within a certain range
2: **if** $mod(t,2) == 0$ **then**
3: $o_2 = \mathcal{U} + \mathcal{L} - o_1$
4: **else**
5: $o_2 = sr \times (\mathcal{U}_k + \mathcal{L}_k) - o_1$
6: **end if**
7: $t = t + 1$.

3) Q3: What are the advantages and innovations of MFEA-DGD compared to other multi-task evolutionary algorithms that introduce gradient approximation methods?

All experiments were carried out on a PC running Windows 10, with an Intel Core i7-8700 CPU running at 3.20GHz and 16GB of RAM.

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 optimum, 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 optimum in the unified express space, i.e., complete, partial, and no intersection (denoted as CI, PI, and NI). More details of functions can be referred to [63], we summarize several properties of these problems in Table III to facilitate our subsequent analysis. Moreover, test suite 2 contains ten MTO problems, taken from the test suite 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

Our experiments are divided into two parts. In the first part, we evaluate the proposed MFEA-DGD method and five representative comparisons on two benchmarks proposed in the CEC 2017 and 2021 EMTO competitions. The compared methods include the original MFEA [1], MFEA-II [20],

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

TABLE III: Summary of properties of problem pairs for evolutionary multitasking.

Category	Task	Global Minimum(θ^*)	Search Space	Degree of Intersection	Distance of Global Minimums
CI+HS	Griewank (T_1)	$(0,\ldots,0)^T$	$[-100,100]^{50}$	Complete intersection	0
	Rastrigin (T_2)	$(0,\ldots,0)^T$	$[-50,50]^{50}$		
CI+MS	Ackley (T_1)	$(0,\ldots,0)^T$	$[-50,50]^{50}$	Complete intersection	0
	Rastrigin (T_2)	$(0,\ldots,0)^T$	$[-50,50]^{50}$		
CI+LS	Ackley (T_1)	$(42.09,\ldots,42.09)^T$	$[-50,50]^{50}$	Complete intersection	0
	Schwefel (T_2)	$(420.9687,\ldots,420.9687)^T$	$[-500,500]^{50}$		
PI+HS	Rastrigin (T_1)	$(0,\ldots,0)^T$	$[-50,50]^{50}$	Partial intersection	1
	Sphere (T_2)	$(0,\ldots,0,20,\ldots,20)^T$	$[-100,100]^{50}$		
PI+MS	Ackley (T_1)	$(0,\ldots,0,1,\ldots,1)^T$	$[-50,50]^{50}$	Partial intersection	0.1
	Rosenbrock (T_2)	$(1,\ldots,1)^T$	$[-50,50]^{50}$		
PI+LS	Ackley (T_1)	$(0,\ldots,0)^T$	$[-50,50]^{50}$	Partial intersection	0
	Weierstrass (T_2)	$(0,\ldots,0)^T$	$[-0.5,0.5]^{25}$		
NI+HS	Rosenbrock (T_1)	$(1,\ldots,1)^T$	$[-50,50]^{50}$	No intersection	0.1414
	Rastrigin (T_2)	$(0,\ldots,0)^T$	$[-50,50]^{50}$		
NI+MS	Griewank (T_1)	$(10,\ldots,10)^T$	$[-100,100]^{50}$	No intersection	0.7071
	Weierstrass (T_2)	$(0,\ldots,0)^T$	$[-0.5,0.5]^{50}$		
NI+LS	Rastrigin (T_1)	$(0,\ldots,0)^T$	$[-50,50]^{50}$	No intersection	5.9524
	Schwefel (T_2)	$(420.9687,\ldots,420.9687)^T$	$[-500,500]^{50}$		

MFEA-AKT [25], MFEA-GHS [21] and MTEA-AD [27]. The parameter settings of the aforementioned algorithms are summarized as follows:

a) The population size in all algorithms is 100.
b) The maximum number of function evaluations (FEs) in our proposal, MFEA, MFEA-II, MFEA-AKT, MFEA-GHS and MTEA-AD is set to $100000 \times n$, where n 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 σ is random selected from set $\{10^{-i}\}_{i=1}^4$ in each generation, $M = 1$, $\gamma = 0.1$, the random matrix $\mathcal{A}$ in each generation is generated in the following form:

$$\mathcal{A} = \frac{1}{2}\begin{pmatrix} 1+\chi & 1-\chi \\ 1-\chi & 1+\chi \end{pmatrix}, \tag{14}$$

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.

In the second part, we compare MFEA-DGD with the existing algorithm MTES [49], which also provides a theoretic analysis about the convergence, and adopts the OpenAI ES to simulate the gradients. Since MTES does not have open source code, we directly follow the experimental parameters and results in [49]. For the proposed MFEA-DGD, the maximum number of calculations to be changed to $250000 \times n$, and the independent number of runs of the experiment is 30. Other parameter settings are the same as in the first part of the experiments.

C. Results and discussions

We will show experimental results and answer Q1-Q3.

Q1: We first consider Q1, and then compare MFEA-DGD with with five state-of-the-art multitasking methods. The comparison results in terms of the mean and standard deviation of the best-achieved function evaluations over 20 runs for each component task in each MTO problem from the two test suites are reported in Table IV and V. The results are used for comparing the six algorithms via Friedman's test at the significant level of 0.05, which reveals the existence of significant differences between the compared algorithms. The average rank of each compared algorithm as the intermediate output of Friedman's test, is reported as *meanrank* in the penultimate row of Table IV and V. The 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.

It can be seen that MFEA-DGD performs exceptionally well on two benchmarks of continuous MTO problems, as shown in Tables IV and V, in terms of the averaged objective value. Next we have some discussion of the compared algorithms. For

TABLE IV: The averaged standard objective value of six compared methods, over 20 independent runs on the single-objective MTO test suite 1. The *meanrank* is obtained via Friedman's test.

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	**1.03E-07±1.10E-08**	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	4.35E-06±2.73E-06	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	**1.00E-07±2.65E-23**	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	1.53E+00±2.77E+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	**7.60E+01±1.42E+02**	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	**1.11E-07±3.41E-08**	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	1.32E-04±3.55E-05	5.28E+00±1.38E+00(−)	2.83E-02±1.21E-02(−)	7.23E-03±7.13E-03(−)	1.61E+02±3.32E+01(−)	**9.25E-07±6.16E-08**(+)
F5:PI+MS	T_1	1.91E+00±3.77E-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	**7.26E-02±1.08E-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	4.23E-06±2.98E-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	1.24E-03±6.05E-04	1.98E+01±2.23E+00(−)	1.09E+01±2.15E+00(−)	2.54E+00±8.34E-01(−)	2.20E-01±1.97E-01(−)	**9.80E-05±1.02E-04** (+)
F7:NI+HS	T_1	**2.61E-02±2.38E-02**	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	**1.00E-07±2.65E-23**	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	1.02E-05±7.11E-06	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	**2.41E-03±1.45E-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	**1.01E-07±4.03E-09**	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	8.45E+03±2.20E+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**(+)
meanrank		1.72	5.67	3.89	4.39	2.94	2.39
−/≈/+			17/0/1	15/1/2	16/1/1	16/1/1	10/2/6

TABLE V: The averaged standard objective value of six compared methods, over 20 independent runs on the single-objective MTO test suite 2. The *meanrank* is obtained via Friedman's test.

Problem	Task	MFEA-DGD	MFEA	MFEA-II	MFEA-AKT	MFEA-GHS	MTEA-AD
1	T_1	6.17E+02±1.62E+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±1.77E+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	7.00E+02±1.20E-02	7.01E+02±1.98E-02(−)	7.00E+02±5.24E-02(−)	7.01E+02±1.14E-01(−)	7.01E+02±7.36E-02(−)	**7.00E+02±2.30E-02**(+)
	T_2	**7.00E+02±8.86E-03**	7.01E+02±1.35E-02(−)	7.00E+02±4.44E-02(−)	7.01E+02±1.08E-01(−)	7.01E+02±6.48E-02(−)	7.00E+02±1.99E-02(−)
3	T_1	**3.71E+04±1.89E+04**	3.62E+06±2.05E+06(−)	1.59E+06±8.80E+05(−)	4.37E+06±2.22E+06(−)	2.55E+05±2.10E+05(−)	5.62E+06±2.36E+06(−)
	T_2	**6.23E+04±3.77E+04**	3.52E+06±1.74E+06(−)	1.90E+06±1.49E+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±5.43E-02**	1.30E+03±1.15E-01(−)	1.30E+03±1.03E-01(−)	1.30E+03±1.11E-01(−)	1.30E+03±6.48E-02(−)	1.30E+03±6.17E-02(−)
	T_2	**1.30E+03±7.19E-02**	1.30E+03±5.58E-02(−)	1.30E+03±5.30E-02(−)	1.30E+03±7.76E-02(≈)	1.30E+03±4.84E-02(−)	1.30E+03±5.49E-02(−)
5	T_1	1.56E+03±2.04E+01	1.56E+03±1.29E+01(≈)	**1.52E+03±4.48E+00**(+)	1.54E+03±7.16E+00(+)	1.54E+03±8.09E+00(+)	1.53E+03±1.44E+00(+)
	T_2	1.55E+03±1.54E+01	1.55E+03±1.01E+01(≈)	**1.52E+03±3.61E+00**(+)	1.54E+03±5.29E+00(≈)	1.54E+03±6.27E+00(≈)	1.53E+03±1.59E+00(+)
6	T_1	**4.15E+05±2.47E+05**	1.73E+06±7.19E+05(−)	1.26E+06±6.92E+05(−)	1.90E+06±1.15E+06(−)	7.42E+05±6.86E+05(≈)	6.91E+06±6.90E+06(−)
	T_2	**2.73E+05±1.93E+05**	1.39E+06±8.37E+05(−)	8.30E+05±3.42E+05(−)	2.30E+06±1.31E+06(−)	4.20E+05±2.67E+05(−)	1.06E+07±6.91E+06(−)
7	T_1	3.16E+03±3.66E+02	3.33E+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	**3.10E+03±2.77E+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±3.91E-02**	5.20E+02±1.09E-01(−)	5.21E+02±3.37E-02(−)	5.21E+02±1.18E-01(−)	5.20E+02±1.11E-01(−)	5.21E+02±3.14E-02(−)
	T_2	**5.20E+02±3.46E-02**	5.20E+02±7.92E-02(−)	5.21E+02±3.47E-02(−)	5.21E+02±1.48E-01(−)	5.20E+02±9.57E-02(−)	5.21E+02±3.46E-02(−)
9	T_1	**8.32E+03±9.15E+02**	8.36E+03±7.04E+02(≈)	8.27E+03±7.19E+03(≈)	8.64E+03±9.74E+02(≈)	8.59E+03±7.69E+02(≈)	1.50E+04±2.45E+02(−)
	T_2	**1.62E+03±7.88E-01**	1.62E+03±4.03E-01(−)	1.62E+03±7.68E-01(−)	1.62E+03±4.95E-01(−)	1.62E+03±7.55E-01(−)	1.62E+03±1.72E-01(−)
10	T_1	**6.22E+03±2.63E+03**	3.62E+04±1.76E+04(−)	2.67E+04±1.18E+04(−)	5.00E+04±1.85E+04(−)	2.25E+03±1.07E+04(−)	5.84E+04±1.56E+04(−)
	T_2	**5.59E+05±2.15E+05**	3.20E+06±1.65E+06(−)	2.12E+06±9.95E+05(−)	3.42E+06±2.00E+06(−)	1.38E+06±1.20E+06(−)	1.77E+07±9.42E+06(−)
meanrank		1.7	4.65	3	4.55	2.95	4.15
−/≈/+			15/5/0	15/3/2	15/4/1	12/7/1	13/2/5

MFEA, negative transfer is unavoidable because knowledge transfer occurs randomly. However, MFEA-DGD simulates the dynamics of the DGD algorithm, which can combine the convexity of different tasks to enhance positive transfer. In some sense, strong enough positive transfer can offset the harm caused by negative transfer. MFEA-II optimizes the probability of knowledge transfer between tasks, its ability to overcome negative transfer is still inferior to MFEA-DGD. 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 15 of 20 tasks, respectively. This demonstrates that, while MFEA-AKT can adaptively select the appropriate crossover operator from SBX, Arithmetical, Geometrical, and BLX-α, the new crossover and mutation operators designed in MFEA-DGD can assist in more effectively searching for the global optimums. Regard to MFEA-GHS, which employs the hyper-rectangle search strategy as well, and the general framework of the algorithm is the same as MFEA-DGD. The comparison results with MFEA-GHS shows that the new operators we proposed outperform the SBX crossover, PM, and genetic transform strategy used in MFEA-GHS, thus reconfirms the advantage of the idea of introducing quasi-gradient descent. MTEA-AD is a very superior algorithm that outperforms many existing EMTO methods, and its framework is different from several other MFEAs. Our proposed MFEA-DGD outperforms MTEA-AD on both two test suites, which indicates that the proposed algorithm is indeed very competitive.

Q2: We now proceed by addressing question Q2. We will explain the experimental results from the theoretical properties of the DGD algorithm implied by Theorem 1.

We depict the average convergence trends of the six compared approaches on all problems of the test suite 1 and 2. Due to the page limit, the figures are provided in the online document http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/trends.pdf. In these figures, the x-axis indicates the number of function evaluations, while the y-axis indicates the average objective value on a log scale. To prevent illegal values on a log scale, the average objective value of a solved task is set to 1E-07 in order to prevent illegal values. As shown in these results, MFEA-DGD has the fastest convergence rate for most tasks. Especially in the early stage, the convergence rate of MFEA-DGD is amazing, so the advantage of MFEA-DGD will be greater than other algorithms in the case of insufficient computational resources. This is in fully consistent with the geometric convergence rate of the DGD algorithm stated in Theorem 1. Since we introduce the idea of quasi-gradient descent, the convergence rate of MFEA-DGD is essentially improved compared with other evolutionary algorithms when the local convexity of the tasks is good.

Next, for the nine problems in test suite 1, we will combine the properties of the functions in Table III and Theorem 1 to explain why MFEA-DGD performs extremely well on some problems, but is less advantageous or even less effective on some other problems. In view of Theorem 1, it follows that for every pair of twin tasks, the DGD algorithm can converge quickly if their global optimums are close and satisfy the joint strong convexity in Assumption 2 (Assumption 1 obviously holds in the setting of test suite 1, and Assumption 3 can be guaranteed by appropriately selecting the hyperparameters). Therefore, since MFEA-DGD can simulate the dynamic equations of DGD, its performance for different problems on test suite 1 is also determined by two key points: the distance between the global optimums of the twin tasks, the strong convexity of the sum of the twin tasks. The former can be seen directly from Table III, the last column of Table III represents the Euclidean distance between the global minimums of every pair of twin tasks on the unified express space $[-1,1]^{50}$. For the latter, we use the following quantity to measure the strong convexity of the function. Given function $f:\Theta\mapsto\mathbb{R}$, we define

$$C(f)=\frac{\mathcal{L}(\{\theta\in\Theta:\nabla^2 f(\theta)\text{ is positive definite}\})}{\mathcal{L}(\Theta)}$$

where Θ is a bounded search space with dimension d. Clearly, the size of $C(f)$ measures the strong convexity of the function f. Based on this definition, we can do a brief review of the strong convexity of the classes of functions used in test suite 1 from from [63]. After simple estimations and calculations (details are available at http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/proof.pdf), we provide the values of $C(f)$ corresponding to each function in Table VI with $\Theta=[-B,B]^d$, which helps us to compare the convexity of the different pairs of twin tasks in test suite 1. It is worth noting that $C(Schwefel)\approx 0.5^d$ in the Table VI implies $\lim_{B\to\infty} C(Schwefel)=0.5^d$, but we cannot tell whether $C(Schwefel)$ is greater than 0.5^d for general B. In fact, the sign of $C(Schwefel)-0.5^d$ is switched alternately as B increases. Now, with the exception of $Schwefel$, for the other six functions we can easily give their convexity ranking on the general search space. Now what we need to do is to verify our analysis above by combining the experiments results.

TABLE VI: Estimations of $C(f)$

Function	C(Function)	B
Sphere	1	$B>0$
Rosenbrock	$>\frac{1}{8B}\left(1-\frac{6}{B}\right)^{d-1}$	$B\geq 6$
Ackley	$<0.496^d$	$B\geq 50$
Rastrigin	0.502^d	$B\in\mathbb{N}^+$
Griewank	$>0.53^d$	$B\geq 50$
Weierstrass	0.5^d	$2B\in\mathbb{N}^+$
Schwefel	$\approx 0.5^d$	$B\to\infty$

According to the level of distance of global minimums, we divide the nine problems in test suite 1 into four subsets for discussion, $\{F_1,F_2,F_3,F_6\}$, $\{F_5,F_7\}$, $\{F_4,F_8\}$, $\{F_9\}$. First, for problem F_1,F_2,F_3,F_6, the global minimums of their corresponding twin tasks are exactly coincident in the unified express space (note that although problem F_5 is PI, this is caused by the different dimensionality of its twin tasks, if task 2 of F_5 is lifted from dimension 25 to 50, it still has the same global minimum as task 1). By the idea of control variables, next we only need to focus on their convexity. Since

$$\begin{aligned}&C(Griewank)+C(Rastrigin)\\>\;&C(Ackley)+C(Rastrigin)\\>\;&C(Ackley)+C(Weierstrass),\end{aligned}$$

the order of the convexity ranking of F_1,F_2,F_6 is exactly the same as MFEA-DGD's ranking of the best fitness above them from smallest to largest, refer to Table IV. Similarly, by

$$\begin{aligned}&C(Griewank)+C(Rastrigin)\\>\;&C(Ackley)+C(Schwefel),\end{aligned}$$

this is consistent with the comparison of the average ojective values of F_1 and F_3 in Table IV. Next, consider problem F_5 and F_7, the distances between their global minimums are 0.1 and 0.1414, respectively, which are close but not exactly equal. Notice that

$$\begin{aligned}&C(Rosenbrock)+C(Rastrigin)\\>\;&C(Ackley)+C(Rosenbrock),\end{aligned}$$

the direction of this inequality and the best fitness of F_5 and F_7 under the MFEA-DGD algorithm are also consistent. Next, we consider problem F_4 and F_8 where the global minimums are further away, with

$$\begin{aligned}&C(Rastrigin)+C(Sphere)\\>\;&C(Griewank)+C(Weierstrass).\end{aligned}$$

Therefore, the convexity of F_4 is better than that of F_8, and according to our theory, MFEA-DGD may performs better on problem F_4, which is fully consistent with the experimental results. Finally, it is easy to find that MFEA-DGD performs the worst on problem F_9, and the reason is well explained

TABLE VII: Objective function value of the test suite 1 under MFEA-DGD and MTES. The results are averaged over 30 runs.

Index	MFEA-DGD		MTES	
	T1	T2	T1	T2
1	$\mathbf{0.00E+0}$	$\mathbf{0.00E+0}$	$1.20\text{E}-3$	$3.08\text{E}+1$
2	$\mathbf{7.51E-6}$	$\mathbf{0.00E+0}$	$3.27\text{E}-1$	$4.80\text{E}+1$
3	$\mathbf{4.76E-4}$	$\mathbf{6.50E-4}$	$2.00\text{E}+1$	$9.49\text{E}+3$
4	$\mathbf{0.00E+0}$	$2.77\text{E}-5$	$3.06\text{E}+1$	$\mathbf{0.00E+0}$
5	$1.55\text{E}+0$	$\mathbf{1.97E-2}$	$\mathbf{5.76E-2}$	$5.04\text{E}+1$
6	$\mathbf{3.35E-6}$	$\mathbf{1.49E-3}$	$2.81\text{E}+0$	$6.63\text{E}+0$
7	$\mathbf{9.63E-3}$	$\mathbf{0.00E+0}$	$4.25\text{E}+1$	$3.07\text{E}+1$
8	$\mathbf{2.33E-6}$	$\mathbf{3.01E-3}$	$9.76\text{E}-2$	$5.78\text{E}+0$
9	$\mathbf{0.00E+0}$	$\mathbf{8.16E+3}$	$4.85\text{E}+3$	$1.33\text{E}+4$

because the distance between the global minimums of the twin tasks in F_9 is much larger than that of the other problems, so theoretically the MFEA-DGD algorithm will not perform well on such a problem.

Q3: At the end of this section, we answer Q3. In fact, the MTES algorithm is the first multi-task evolutionary algorithm in the existing literature that simultaneously establishes a strict convergence analysis with the gradient descent approximation. Compared to MTES, the main advantage of our proposed MFEA-DGD is the combination of the MFEA framework and gradient descent, which makes the algorithm much more scalable and enhances the diversity of populations. Furthermore, our convergence analysis does not assume that each task is a convex function, which is more realistic. In Table VII, we compare the performance of the two algorithms on test suite 1. It is clear that MFEA-DGD significantly outperforms MTES on almost all tasks with excellent performance.

VI. Conclusions

In this paper, we theoretically prove that the DGD method can effectively overcome the non-convex optimization task and has the property of fast convergence. Moreover, we propose a MFEA-DGD algorithm that extends MFEA by combining new reproduction operators and a hyper-rectangle search strategy. The MFEA-DGD is characterized by two novel crossover and mutation operators, which simulate the dynamics of the DGD algorithm, and the OpenAI ES is used to estimate the unknown gradient. The interpretability 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, the hyper-rectangle strategy is used to broaden the algorithm's search range. On two MTO test suites, we compare MFEA-DGD to some classical or new EMTO algorithms to demonstrate its superiority. Furthermore, we provide a theoretical explanation for the experimental results. 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.

References

[1] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2016.
[2] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cognitive Computation*, vol. 8, no. 2, pp. 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*, vol. 47, no. 3, pp. 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*, vol. 21, no. 5, pp. 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*, pp. 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)*, pp. 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)*, pp. 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*, vol. 243, pp. 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*, vol. 50, no. 4, pp. 1–26, 2020.
[10] L. Feng, Y.-S. Ong, M. H. Lim, and I. W. Tsang, "Memetic search with interdomain learning: A realization between cvrp and carp," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 644–658, 2015.
[11] L. Feng, L. Zhou, A. Gupta, J. Zhong, Z. Zhu, K. C. Tan, and K. Qin, "Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3171–3184, 2021.
[12] H. Li, Y.-S. Ong, M. Gong, and Z. Wang, "Evolutionary multitasking sparse reconstruction: Framework and case study," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 733–747, 2019.
[13] N. Zhang, A. Gupta, Z. Chen, and Y.-S. Ong, "Evolutionary machine learning with minions: A case study in feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 130–144, 2022.
[14] D. Wu and X. Tan, "Multitasking genetic algorithm (MTGA) for fuzzy system optimization," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 6, pp. 1050–1061, 2020.
[15] T. Wei, S. Wang, J. Zhong, D. Liu, and J. Zhang, "A review on evolutionary multi-task optimization: Trends and challenges," *IEEE Transactions on Evolutionary Computation (Early Access)*, pp. 1–20, 2021.
[16] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization - a new frontier in evolutionary computation research," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 22–33, 2021.
[17] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and T. P. Siew, "Linearized domain adaptation in evolutionary multitasking," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 1295–1302.
[18] Y.-W. Wen and C.-K. Ting, "Parting ways and reallocating resources in evolutionary multitasking," *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2404–2411, 2017.
[19] J. Ding, C. Yang, Y. Jin, and Y. Chai, "Generalized multi-tasking for evolutionary optimization of expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 44–58, 2019.
[20] 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*, vol. 24, no. 1, pp. 69–83, 2019.

[21] 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*, vol. 138, p. 112798, 2019.

[22] X. Zheng, K. Qin, M. Gong, and D. Zhou, "Self-regulated evolutionary multitask optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 16–28, 2020.

[23] M. Gong, Z. Tang, H. Li, and J. Zhang, "Evolutionary multitasking with dynamic resource allocating strategy," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 858–869, 2019.

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

[25] 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*, vol. 51, no. 5, pp. 2563–2576, 2020.

[26] Z. Tang, M. Gong, Y. Wu, W. Liu, and Y. Xie, "Regularized evolutionary multitask optimization: Learning to intertask transfer in aligned subspace," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 262–276, 2021.

[27] C. Wang, J. Liu, K. Wu, and Z. Wu, "Solving multi-task optimization problems with adaptive knowledge transfer via anomaly detection," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 304–318, 2022.

[28] Z. Tang, M. Gong, Y. Wu, A. K. Qin, and K. C. Tan, "A multifactorial optimization framework based on adaptive intertask coordinate system," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6745–6758, 2022.

[29] J.-Y. Li, Z.-H. Zhan, K. C. Tan, and J. Zhang, "A meta-knowledge transfer-based differential evolution for multitask optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 719–734, 2021.

[30] H. Han, X. Bai, H. Han, Y. Hou, and J. Qiao, "Self-adjusting multi-task particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 145–158, 2021.

[31] X. Ma, J. Yin, A. Zhu, X. Li, Y. Yu, L. Wang, Y. Qi, and Z. Zhu, "Enhanced multifactorial evolutionary algorithm with meme helper-tasks," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7837–7851, 2022.

[32] 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*, vol. 49, no. 9, pp. 3457–3470, 2018.

[33] Z. Liang, X. Xu, L. Liu, Y. Tu, and Z. Zhu, "Evolutionary many-task optimization based on multisource knowledge transfer," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 319–333, 2022.

[34] R.-T. Liaw and C.-K. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 2266–2273.

[35] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 369–384, 2019.

[36] J. Tang, Y. Chen, Z. Deng, Y. Xiang, and C. P. Joy, "A group-based approach to improve multifactorial evolutionary algorithm." in *IJCAI*, 2018, pp. 3870–3876.

[37] S. Huang, J. Zhong, and W.-J. Yu, "Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization," *IEEE transactions on emerging topics in computing*, vol. 9, no. 4, pp. 1930–1944, 2019.

[38] H. Xu, A. K. Qin, and S. Xia, "Evolutionary multitask optimization with adaptive knowledge transfer," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 290–303, 2021.

[39] X. Xue, K. Zhang, K. C. Tan, L. Feng, J. Wang, G. Chen, X. Zhao, L. Zhang, and J. Yao, "Affine transformation-enhanced multifactorial optimization for heterogeneous problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6217–6231, 2022.

[40] Q. Shang, L. Zhang, L. Feng, Y. Hou, J. Zhong, A. Gupta, K. C. Tan, and H.-L. Liu, "A preliminary study of adaptive task selection in explicit evolutionary many-tasking," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 2153–2159.

[41] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE transactions on cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.

[42] J. Lin, H.-L. Liu, B. Xue, M. Zhang, and F. Gu, "Multiobjective multitasking optimization based on incremental learning," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 824–838, 2019.

[43] 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*, vol. 51, no. 4, pp. 1784–1796, 2020.

[44] Z. Liang, W. Liang, Z. Wang, X. Ma, L. Liu, and Z. Zhu, "Multiobjective evolutionary multitasking with two-stage adaptive knowledge transfer based on population distribution," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.

[45] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, "Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution," *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 2096–2109, 2022.

[46] Z. Chen, Y. Zhou, X. He, and J. Zhang, "Learning task relationships in evolutionary multitasking for multiobjective continuous optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5278–5289, 2022.

[47] J. Lin, H.-L. Liu, K. C. Tan, and F. Gu, "An effective knowledge transfer approach for multiobjective multitasking optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3238–3248, 2021.

[48] H. Chen, H.-L. Liu, F. Gu, and K. C. Tan, "A multi-objective multitask optimization algorithm using transfer rank," *IEEE Transactions on Evolutionary Computation*, 2022.

[49] 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*, vol. 52, no. 8, pp. 145–158, 2021.

[50] H. Han, X. Bai, Y. Hou, and J. Qiao, "Self-adjusting multi-task particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 145–158, 2021.

[51] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[52] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[53] 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.

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

[55] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2009.

[56] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

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

[58] S. Xie and L. Guo, "A necessary and sufficient condition for stability of LMS-based consensus adaptive filters," *Automatica*, vol. 93, pp. 12–19, 2018.

[59] G. Pavai and T. Geetha, "A survey on crossover operators," *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, pp. 1–43, 2016.

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

[61] X. Qiu, K. C. Tan, and J.-X. Xu, "Multiple exponential recombination for differential evolution," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 995–1006, 2017.

[62] Y. Nesterov, *Lectures on convex optimization*. Berlin: Springer International Publishing, 2018.

[63] 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.