Multifactorial Evolutionary Algorithm Based on Diffusion Gradient Descent

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

***Abstract*—The multifactorial evolutionary algorithm (MFEA) is one of the most widely used evolutionary multitasking algorithms. The MFEA implements knowledge transfer among optimization tasks via crossover and mutation operators and it obtains high-quality solutions more efficiently than 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 demonstrate that the local convexity of some tasks can help other tasks escape from local optima via knowledge transfer. Based on this theoretical foundation, we design complementary crossover and mutation operators for the proposed MFEA-DGD. As a result, the evolution population is endowed with a dynamic equation that is similar to DGD, i.e., convergence is guaranteed, and the benefit from knowledge transfer is explainable. In addition, a hyper-rectangular search strategy is introduced to allow MFEA-DGD to explore more underdeveloped areas in the unified express space of all tasks and the subspace of each task. The proposed MFEA-DGD is verified experimentally on various multitask optimization problems, and the results demonstrate that MFEA-DGD can converge faster to competitive results compared to state-of-the-art evolutionary multitasking algorithms. We also show the possibility of interpreting the experimental results based on the convexity of different tasks.**

***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 (EA). Traditional EAs only solve a single optimization task; however, many real-world optimization tasks are related, and valuable knowledge gained from solving one task can help solve another related task [3], [4]. By exploiting the advantages of knowledge transfer, it has been demonstrated that EMT can outperform single-task EAs on various optimization problems [5]–[9] and real-world applications [10]–[14].

The MFEA [1] has received rapidly increasing attention due to its simplicity and efficiency, and numerous improvements and MFEA variants have been surveyed in the literature [15], [16]. The main concepts of some representative algorithms [17]–[33] are summarized in Table I. In addition, the MFEA has been extended to solve many-task problems that, which involve more than two optimization tasks [34]–[43], and multi-objective multitask problems [44]–[53]. .

TABLE I: Representative EMT algorithms.

Algorithm	Main Ideas
LDA-MFEA [17]	Linearized domain adaptation strategy
MFEARR [18]	Detection of parting ways and reallocating fitness evaluations
G-MFEA [19]	Decision variable translation and shuffling
MFEA-II [20]	Online random mating parameter learning
MFEA-GHS [21]	Genetic transform and hyper-rectangle search
SREMTO [22]	Self-regulated knowledge transfer
MTO-DRA [23]	Online dynamic resource allocation
MFMP [24]	Genetic transfer based on multi-population
MFEA-AKT [25]	Adaptive knowledge transfer
ASCMFDE [26]	Inter-task knowledge transfer in low-dimension subspaces
MTEA-AD [27]	Anomaly detection model
GFMFDE [28]	Knowledge transfer based on active coordinate system
MKTDE [29]	Meta knowledge transfer
SA-MTPSO [30]	Knowledge estimation metric and self-adjusting knowledge transfer
MFEA/MVD [31]	Multi-objective decomposition based helper-task
DKT-MTPSO [32]	Diversified knowledge transfer
DAMTO [33]	Domain adaptation-based mapping strategy

MFEAs have been applied successfully to various complex optimization problems. However, to the best of our knowledge, strict theoretical analyses of the convergence of MFEAs and the benefits of knowledge transfer are scarce. Bali *et al.* [20] presented a pioneering attempt to prove the algorithm's convergence and analyze the effects of inter-task interactions in MFEA using a probability distribution to model the target population. Here, the convergence of the probability distribution requires updating the probability density at each point in the search space, which strictly implies that the entire space must be searched with an infinite-sized population. However, this underlying assumption may be unrealistic in MFEAs. For other EMT algorithms, Bali *et al.* [54] also presented a convergence-guaranteed multitask gradient descent (GD) algorithm; however, although the convergence proof was effective for convex problems, it may not hold for nonconvex tasks, which are more prevalent in practical applications. In addition, Han *et al.* [55] performed a convergence analysis of a particle swarm optimization-based EMT algorithm, which suffers from the same issue encountered by [20].

Thus, in this paper, we propose an MFEA based on diffusion gradient descent (DGD). The proposed MFEA-DGD enables a more general theoretical explanation of knowledge transfer and

This work was supported in part by the National Natural Science Foundation of China, under Grant 61871272, the Guangdong Provincial Key Laboratory, under Grant 2020B121201001, Guangdong Basic and Applied Basic Research Foundation, under Grant 2021A1515011911, and Shenzhen Science and Technology Program, under Grant JCYJ20220531102617039, 20200811181752003 (Corresponding author: Zexuan Zhu.)

Z. Liu is with the College of Computer Science and Software Engineering, and the Institute for Advanced Study, Shenzhen University, Shenzhen 518060, China. (e-mail:liuzhaobo@szu.edu.cn)

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

H. Zhang is with the Institute of Applied Mathematics, Shenzhen Polytechnic, Shenzhen 518055, China. (e-mail:zhanghl@szpt.edu.cn)

population convergence. We initially describe the motivation to use DGD, and then we theoretically prove the validity of knowledge transfer and fast convergence in DGD for multiple similar optimization problems (including nonconvex tasks). Based on DGD, we design new crossover and mutation operators to replace the simulated binary crossover (SBX) [56] and polynomial mutation (PM) [57] operations utilized in the traditional MFEA. By introducing these new operators, the proposed MFEA-DGD can approximate the dynamic equation of DGD, thereby endowing it with population convergence and theoretical interpretability of knowledge transfer. The analytic form of the gradient of the optimization functions is not available directly (or is nonexistent); thus, we utilized the estimation method in OpenAI evolutionary strategy (ES) [58], [59] to simulate the gradient. In addition, a hyper-rectangular search strategy [21] is introduced to the proposed MFEA-DGD to enable exploration of more undeveloped areas. The effectiveness of the proposed MFEA-DGD is demonstrated through experiments involving multitask optimization (MTO) problems, in which it outperforms state-of-the-art MFEAs and EMT algorithms. To further verify the consistency between the experimental and theoretical results, we estimate the convexity of each pair of twin tasks in the benchmark and calculate the distance between their global optima.

Our primary contributions are summarized as follows.

- We present a groundbreaking theoretical demonstration of the convergence of the DGD algorithm for multitasking. This is a significant departure from existing results in the literature about nonconvex optimization algorithms because it employs a novel approach of transferring gradient information to optimize multiple nonconvex tasks. Specifically, we introduce joint strong convexity as a new condition for the convergence of the DGD algorithm, and we demonstrate that, when the optimal solutions for different tasks are close in distance, DGD proves to be an effective method, achieving geometric convergence toward a neighborhood of the global optima.
- We introduce significant new features and advancements by incorporating novel crossover and mutation operators into MFEA based on DGD. This results in the proposed MFEA-DGD. This unique approach enables the algorithm to simulate the optimization process of DGD, thereby demonstrating how these innovative crossover and mutation operators contribute to the enhanced performance of algorithms for similar tasks. The proposed MFEA-DGD is an effective example of the fusion of decentralized optimization algorithms and MFEA, and it demonstrates strong interpretability and rapid convergence.
- Diverging from traditional analyses of experimental results based on task relevance, we present an innovative approach that introduces the task convexity concept. This perspective allows for deeper understanding and more effective interpretation of the experimental results, and it allows us to highlight the true value and originality of the research presented in this paper.

The remainder of this paper is organized as follows. Section II introduces some preliminaries regarding MFEA and DGD. Section III presents a theoretical analysis of DGD and the design principles of the operators. Section IV describes the proposed MFEA-DGD method in detail, and Section V describes empirical experiments conducted to evaluated the performance of the proposed method. Finally, Section VI concludes the paper. Table II summarizes the symbols utilized in this paper.

TABLE II: Notations.

$\mathbf{1}_k$	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 $\lvert a\rvert \leq C \cdot b$.

II. PRELIMINARIES

In this section, we present preliminaries for the methodologies utilized in the proposed MFEA-DGD. First, we introduce the conventional MFEA, and then we demonstrate the principle of the DGD algorithm, which together with OpenAI ES derives the crossover and mutation operators in the proposed MFEA-DGD.

A. Multifactorial evolutionary algorithm

Without loss of generality, an 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. The 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 relative 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 these definitions, the framework of the MFEA is outlined in Algorithm 1. In the initial stage, the population comprises N individuals generated randomly in a unified express space. Then, each individual is randomly assigned a

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 φ and skill factor τ 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 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

skill factor and evaluated in terms of the factorial cost. In each generation of evolution, assortative mating and vertical cultural transmission mechanisms are employed to reproduce offspring through crossover and mutation operators. The exchange of genetic information between individuals facilitates knowledge sharing between different tasks. The vertical cultural transmission mechanism enables individuals with different skill factors to mate with a certain probability, thereby optimizing each task. After the offspring population is generated, the factorial cost, factorial rank, scalar fitness, and skill factor of each individual are updated. Then, elite-based environmental selection is applied to generate a new population from the union of the parent and offspring populations. This evolution procedure is repeated until a stopping criterion is met.

B. Diffusion gradient descent

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

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

where f_i can be nonconvex. Without out loss of generality, we assume $\theta_i \in \mathbb{R}^d$. We begin by considering a case with exact gradients such that GD can be implemented. Here, 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 an efficient algorithm; however, for nonconvex problems, GD tends to get stuck in local optima. We consider diffusion strategies [60] on GD to overcome this limitation by employing useful information between similar tasks. These diffusion strategies are beneficial compared to purely noncooperative strategies provided that the local optima are sufficiently close to each other [61].

A typical version of DGD (Algorithm 2) was used in this study. Here let $\theta_{t,i}$ denote the estimate of the minimizer of task i and time instant t. Similar to the diffusion LMS [61], the general structure of the DGD algorithm involves the following steps:

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

The nonnegative 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 $\mathcal{C}$:

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

Note that appropriate selection of $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{C}$ yields several adaptive strategies as special cases of (2). With the setting $\mathcal{A}_1 = I_n$, we obtain the adapt-then-combine (ATC) DGD. In addition, setting $\mathcal{A}_2 = I_n$ yields the combine-then-adapt (CTA) DGD, and by setting $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{C} = I_n$, the algorithm reverts to standard GD without knowledge transfer. According to a previous study [60], the ATC diffusion LMS algorithm tends to outperform the CTA version of the algorithm. Note that LMS is essentially a GD of a quadratic optimization problem; thus, in this study, we utilized the ATC version of DGD. To facilitate the follow-up study, we consider a common case [62] with $\mathcal{C} = I_n$, where (2) becomes

$$\theta_{t+1,i} = \sum_{j=1}^{n} a_{ij}(\theta_{t,j} - \eta \nabla f_j(\theta_{t,j})), \quad i = 1, \ldots, n, \quad (3)$$

and $\mathcal{A}_1 = \mathcal{A}_2 = \{a_{ij}\}_{n \times n}$.

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

In this section, we first introduce a novel theoretical result demonstrating the suitability of DGD to solve nonconvex optimization problems. We demonstrate that the gradient information of each task can be combined effectively to achieve fast convergence. We then introduce the proposed crossover and mutation operators, which are inspired by DGD with the gradient simulated using OpenAI ES.

A. Convergence of DGD

Differing from existing studies, we demonstrate 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

$$f^{glob} \triangleq \sum_{i=1}^{n} f_i.$$

We give several definitions prior to describing the corresponding theorem.

Definition 1. *A 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 referred to as 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* $i_1, \ldots, i_l$ *that contains* $\{1, \ldots, n\}$, *satisfies* $m_{i_k i_{k+1}} > 0$, *where* $i_{l+1} = i_1$.

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

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

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

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

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

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

Assumption 2. *There exists a 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.*

Remark 1. *Assumption 1 is a conventional condition in nonconvex optimization theory. Assumption 2 represents joint strong convexity in some sense, thereby allowing for multiple nonconvex tasks among* n *tasks. Assumption 2 can be satisfied easily 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

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

From the above, we obtain the following theorem.

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

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

The proof of Theorem 1 is provided at http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/proof.pdf. Here, if $\theta_1^*, \ldots, \theta_n^*$ are 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 other words, under the evolutionary strategy* (3), *the similarity of the global optima 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 optima.*

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

Remark 4. *Note that matrix* $\mathcal{A}$ *in Theorem 1 can be replaced by the time-varying* $N \times N$ *matrix* $\mathcal{A}_t = \{a_{t,li}\}$, *and the gradient of functions* $\nabla f_1, \ldots, \nabla f_n$ *can also be replaced by 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 our EA is heavily dependent on this case. In this case, a similar conclusion to Theorem 1 can still be established, under the assumption that there is an integer* $m > 0$ *such that* $\sum_{j=t}^{t+m} \sum_{i=1}^N \nabla^2 f_{j,i} \geq \xi I_d$, *where each* $B_t = \mathcal{A}_{t+m} \mathcal{A}_{t+m-1} \cdots \mathcal{A}_t$ *satisfies Assumption 3, and the nonzero entries of matrices* $\mathcal{A}_t$ *have the uniform lower bound* a:

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

The proof process is nearly the same; thus, the corresponding proof is omitted. In addition, interested readers can refer to Assumption 6 in the literature [63], which is very similar to our assumptions here.

B. New crossover and mutation operators

DGD has fast convergence and interpretability, which inspires us to design new DGD-based crossover and mutation operators. Generally, crossover is a reproduction operator to exchange genetic materials between parents and generate offspring in EAs. In the last few decades, a large number of crossover operators have been proposed for a wide range of optimization problems [64]–[66]. In the following, we first focus on two typical crossover operators, i.e., the arithmetical and SBX crossover operators, where the element at position i of the offspring is the linear combination of the two selected parents:

$$\begin{cases} c_1^i = \nu \cdot p_1^i + (1 - \nu) \cdot p_2^i \\ c_2^i = (1 - \nu) \cdot p_1^i + \nu \cdot p_2^i \end{cases}. \tag{4}$$

The only difference between these two crossover operators is whether v is selected randomly. Equation (4) can be rewritten in the matrix multiplication form as follows:

$$\mathbf{c} = \mathcal{A} \cdot \mathbf{p}. \tag{5}$$

Here, if the population size is set to $N = 2$, we iteratively generate new offspring according to such a crossover operator. Consider an ideal case where the offspring generated in each step have better fitness than their parents. Then, the population of generation t can by represented as follows:

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

However, such a formula cannot guarantee the convergence rate of the population to the global optima, i.e., it will only produce linear transformations between individuals. Thus, inspired by the DGD algorithm and Theorem 1, is implemented to the above crossover operator (5):

$$\mathbf{c} = \mathcal{A} \cdot \mathbf{p}' \approx \mathcal{A}(\mathbf{p} - \eta \nabla \mathbf{f}(\mathbf{p})), \tag{6}$$

where $\mathbf{f}$ is a two-dimensional vector-valued function, where each component belongs to the set $\{f_1, \ldots, f_n\}$, and the gradient $\nabla \mathbf{f}$ can be approximated using an enhanced OpenAI ES [59].

In the literature, the common PM operator is used to produce offspring by randomly mutating the parents. Note that natural evolution is influenced by various environmental factors and is not entirely random; thus, we expect that the mutation will have directionality. In addition, parental mutations require empirical information from other individuals. Theoretically, the mutation operator was designed using GD, and we expect that the offspring $\mathbf{c}$ will be produced in a quasi-GD direction:

$$\mathbf{c} \approx \mathbf{p} - \eta \nabla \mathbf{f}(\mathbf{p}), \tag{7}$$

which is a special case of (6) with $\mathcal{A} = I_2$.

We apply the crossover (6) and mutation (7) to a population P_{t-1} of N individuals. Assuming that in the given $N/2$ pairs of parents, $N_1/2$ pairs use crossover operators and the remaining $(N - N_1)/2$ pairs use mutation operators. The iteration of the entire population must satisfy the following:

$$P_t \approx \mathcal{A}_{t-1}(P_{t-1} - \eta \nabla \mathbf{f}_{t-1}(P_{t-1})), \tag{8}$$

where

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

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 (8) is completely consistent with the recursive equation of the DGD algorithm (Section II-B); thus, if the crossover and mutation operators are designed according to (6) and (7), respectively, the advantage of DGD can be retained by selecting proper $\mathcal{A}_i, i \geq 0$ (as described in Remark 4). Compared to traditional operators, this strategy realizes a faster convergence rate and can handle the optimization of nonconvex tasks. In addition, the knowledge transfer and convergence of solutions caused by the 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 ..., N/2 do
    Let learning late η = 1/L
    Randomly select two parent individuals p_1, p_2
    Randomly generate a matrix A ∈ R^{2×2}
    Obtain skill factor τ_1, τ_2 of p_1, p_2, respectively
    Let {ξ_j^i}_{j=1}^M be marginally distributed as N(0, I_d),
      i = 1, 2
    Obtain individuals p_{i,-1}^j = p_i − σξ_j^i and p_{i,1}^j = p_i +
      σξ_j^i for each p_i, j = 1, ..., M
    if τ_1 = τ_2 or rand < rmp then
      o_1 = GradTransform(p_1, p_2, ξ^1, ξ^2, A)
      o_2 = Hyper − rectangleSearch(o_1)
    else
      for i = 1, 2 do
        o_i = Quasi − GD(p_i, ξ^i)
      end for
    end if
    for k = 1, ..., 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 ∪ 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, here, we propose the MFEA-DGD based on the new crossover and mutation operators, which enable the algorithm to simulate the optimization process of DGD. The pseudocode for the proposed MFEA-DGD is presented in Algorithm 3, where the functions $Quasi - GD()$, $GradTransform()$ and $Hyper - rectangleSearch()$ are defined in Algorithms 4-6, respectively. Generally, the main difference between the proposed MFEA-DGD and conventional MFEA algorithms lies in the reproduction operators and the hyper-rectangle search strategy. As shown in Algorithm 3, the workflow of the proposed MFEA-DGD can be summarized as follows.

1) Initially, the MFEA-DGD performs the same initialization as the MFEA to generate a population.
2) In each evolutionary generation, two parent individuals, denoted as p_1 and p_2, are selected randomly. 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 randomly to simulate the direction of the GD of f_{τ_i} at p_i.

3) The selected parent individuals p_1 and p_2 mate according to the assorting mating mechanism, which includes the gradient transform strategy. Here, if parents 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. In contrast, if 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 a quasi-GD mutation operator. In this process, 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 rmp value, 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. Thus, the corresponding rmp is small.
4) Finally, after generating and evaluating the offspring population, the elite-based environmental selection operator is employed 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.

Note that the hyper-rectangle search strategy is not our original method (Algorithm 6). This strategy is utilized to expand the search area of the population and increase the diversity of the population; thus, this part is independent from the new operators. In fact, better strategies may be available to enhance the diversity of the population; thus, the proposed algorithm has great potential for improvement.

In a single generation, the proposed MFEA-DGD involves three main steps: gradient transform, quasi-GD mutation, and hyper-rectangle search. Each step takes $O(Nd)$ time, where N and d are the total size of the population and the max number of decision variables in all tasks, respectively. In addition, an elitism-based parameter adaptation strategy has a time complexity of $O(N\log(N/n))$ (e.g., fast sorting), where n is the number of tasks. In summary, the total computational complexity of the proposed MFEA-DGD in a single generation is $O(N(d+\log(N/n)))$.

In the following subsections, we describe the critical components of the proposed MFEA-DGD in detail, including the gradient transform, the quasi-GD mutation, and the criteria used to update the learning rate.

B. Quasi-GD mutation

In contrast to conventional mutation, our mutation criteria are designed by approximating GD with a set number of function evaluations, which is consistent with our intuitive understanding of evolutionary processes because it can be interpreted as individuals selecting the appropriate evolutionary direction based on the experiences of randomly selected individuals in their vicinity. The corresponding pseudocode is given in Algorithm 4, where $\{\epsilon_j\}_{j=1}^M$ are generated marginally by a standard Gaussian distribution $N(0, I_d)$, and $\nabla f_\sigma(p)$ indicates the level of mutation at p. When there is a large

Algorithm 4 The Quasi-GD 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: $p_3 = a_{11}(p_1 - \eta\nabla f_{\sigma,1}(p_1)) + a_{12}(p_2 - \eta\nabla f_{\sigma,2}(p_2))$

difference in fitness between two individuals $p + \sigma\xi_j$ and $p - \sigma\xi_j$, the mutation direction is likely to reduce p's fitness quickly, with p gaining more empirical information from the 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.

C. Gradient transform strategy

During evolution, most offspring are produced by parents who excel at different tasks. However, if the parents are skilled at significantly unrelated tasks, the offspring may not be well-adapted to either task. As a result, it will be difficult for these offspring to survive to the next generation, which will reduce the efficiency of knowledge transfer between tasks. To address this issue, we propose a gradient transform strategy, which is presented in Algorithm 5.

Here, we consider an example two-task problem (Fig. 1) to illustrate the motivation of the proposed gradient transform strategy. Given two parent individuals p_1 and p_2, two intermediate offspring $\tilde{p}_1$ and $\tilde{p}_2$ are produced by applying quasi-GD to them. In addition, a third offspring p_3 is produced by combining these two intermediate offspring in a linear manner (i.e., crossover). As shown in Fig. 1, the Ackley function is nonconvex, and $\tilde{p}_1$, which is obtained after using quasi-GD for p_1, lies at the local optimum of Ackley. Although $\tilde{p}_1$ has better fitness than p_1, $\tilde{p}_1$ is distant from the global optimum of Ackley. In contrast, the Sphere function is convex; thus, using quasi-GD on p_2 (i.e., to produce $\tilde{p}_2$) can approximate the global optimum of the Sphere function effectively. Note that the global optimum of the Ackley and Sphere functions are in similar locations; thus, 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 proposed strategy suggests using a GD direction that converges to the global optima to mitigate the effects of slow GD or misdirection. By exchanging gradient information between different tasks, this strategy can move away from the local optima and converge to the global optima.

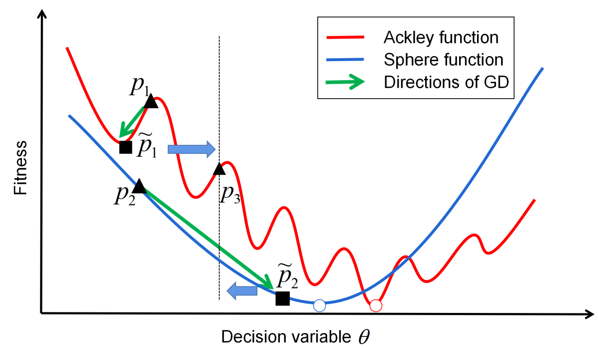

Fig. 1: Proposed gradient transform strategy, where triangles and squares represent the parents and the solution obtained by the parents using GD, respectively. The red and blue lines represent the Ackley and Sphere curves, respectively. Global optima are assumed to be located in the circles.

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: $\quad o_2 = \mathcal{U} + \mathcal{L} - o_1$
4: **else**
5: $\quad o_2 = sr \times (\mathcal{U}_k + \mathcal{L}_k) - o_1$
6: **end if**
7: $t = t + 1$.

D. Learning rate

Here, we consider adaptive selection of the learning rate n. Rather than using a fixed learning rate or a predetermined schedule, the geometry of the target function is utilized to derive the learning rate in an adaptive manner. 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 $\nabla f_{\mathcal{T}_i}$ as follows:

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

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

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

where $\gamma \in (0, 1)$ is a tunable parameter. As each generation of population is renewed, we obtain a new η. The update criteria primarily account for the fact that the best learning rate for GD is typically equal to the Lipschitz constant of the function's gradient [67, p.29].

V. Experiments

In previous sections, we stated that the proposed MFEA-DGD inherits the advantages of DGD and is highly interpretable to effectively solve multitask nonconvex optimization problems. To verify these claims, this section discusses experiments designed to address the following questions.

1) Q1: Is the performance of the proposed MFEA-DGD comparable to that of state-of-the-art EMT algorithms?
2) Q2: Can the performance of the proposed MFEA-DGD be explained by theory?
3) Q3: What are the advantages and innovations of the proposed MFEA-DGD compared to existing multitask EAs that introduce gradient approximation methods?

In this study, all experiments were conducted on a PC running Windows 10 with an Intel Core i7-8700 CPU running at 3.20 GHz with 16 GB of RAM.

A. Test problems

Two suites of test problems were used in these experiments. The first test suite includes nine MTO problems from the CEC 2017 Evolutionary Multitask 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, both of the tasks in the MTO problem are characterized by high, medium, and low similarities (denoted HS, MS, and LS, respectively), and they are classified into three categories based on the degree of intersection of their global optima in the unified express space, i.e., complete, partial, and no intersection (denoted CI, PI, and NI, respectively). Additional details about these functions can be found in the literature [68]. Several properties of the corresponding problems are summarized in Table III to facilitate our subsequent analysis. In addition, the second test suite contains 10 MTO problems taken from the test suite used in the CEC 2021 Evolutionary Multitask Optimization Competition*, where each problem comprises two distinct single-objective optimization tasks, which bear certain commonality and complementarity in terms of the global optimum and the fitness landscape. Note that these MTO problems possess different degrees of latent synergy between their involved component tasks.

B. Experimental settings

Our experiments were divided into two parts. In the first part, we evaluated 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], MFEA-AKT [25], MFEA-GHS [21] and MTEA-AD [27]. In all algorithms, the population size and independent number of runs were set to 100 and 20, respectively. The maximum number of function evaluations (FEs) was set to $100000 \times n$

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

TABLE III: 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)	$(0,\dots,0)^T$	$[-100,100]^{50}$	Complete intersection	0
	Rastrigin (T_2)	$(0,\dots,0)^T$	$[-50,50]^{50}$		
CI+MS	Ackley (T_1)	$(0,\dots,0)^T$	$[-50,50]^{50}$	Complete intersection	0
	Rastrigin (T_2)	$(0,\dots,0)^T$	$[-50,50]^{50}$		
CI+LS	Ackley (T_1)	$(42.09,\dots,42.09)^T$	$[-50,50]^{50}$	Complete intersection	0
	Schwefel (T_2)	$(420.9687,\dots,420.9687)^T$	$[-500,500]^{50}$		
PI+HS	Rastrigin (T_1)	$(0,\dots,0)^T$	$[-50,50]^{50}$	Partial intersection	1
	Sphere (T_2)	$(0,\dots,0,20,\dots,20)^T$	$[-100,100]^{50}$		
PI+MS	Ackley (T_1)	$(0,\dots,0,1,\dots,1)^T$	$[-50,50]^{50}$	Partial intersection	0.1
	Rosenbrock (T_2)	$(1,\dots,1)^T$	$[-50,50]^{50}$		
PI+LS	Ackley (T_1)	$(0,\dots,0)^T$	$[-50,50]^{50}$	Partial intersection	0
	Weierstrass (T_2)	$(0,\dots,0)^T$	$[-0.5,0.5]^{25}$		
NI+HS	Rosenbrock (T_1)	$(1,\dots,1)^T$	$[-50,50]^{50}$	No intersection	0.1414
	Rastrigin (T_2)	$(0,\dots,0)^T$	$[-50,50]^{50}$		
NI+MS	Griewank (T_1)	$(10,\dots,10)^T$	$[-100,100]^{50}$	No intersection	0.7071
	Weierstrass (T_2)	$(0,\dots,0)^T$	$[-0.5,0.5]^{50}$		
NI+LS	Rastrigin (T_1)	$(0,\dots,0)^T$	$[-50,50]^{50}$	No intersection	5.9524
	Schwefel (T_2)	$(420.9687,\dots,420.9687)^T$	$[-500,500]^{50}$		

TABLE IV: Parameter settings for the six compared algorithms.

Algorithm	SBX (p_c, η_c)	PM (p_m, η_m)	rmp	Other Parameters
MFEA	(1,15)	(1/d,15)	0.3	–
MFEA-II	(1,15)	(1/d,15)	–	–
MFEA-AKT	(1,2)	(1/d,20)	0.5	Parameters in Arithmetical, Geometrical, BLX-α Crossover: $\lambda=2$, $\varpi=0.25$, $\alpha=0.03$
MFEA-GHS	(1,2)	(1/d,5)	0.7	$sr\in[0.5,1.5]$, number of top individuals used to calculate the mapping vectors: $n_0=2$
META-AD	–	–	–	Parameter α_0 controls the frequency of knowledge transfer was set to 0.1
MFEA-DGD	–	–	0.7	$\sigma\in\{10^{-i}\}_{i=1}^4$, $M=1$, $\gamma=0.1$, $\mathcal{A}=\frac{1}{2}\begin{pmatrix}1+\chi & 1-\chi\\ 1-\chi & 1+\chi\end{pmatrix}$, $sr\in[0.5,1.5]$

for all methods, where n is the number of tasks. The parameter settings of the aforementioned algorithms are summarized in Table IV. We point out that the scaling rate $sr\in[0.5,1.5]$ means the value of sr was generated randomly within the range of $[0.5,1.5]$, smoothing parameter $\sigma\in\{10^{-i}\}_{i=1}^4$ means that σ was random selected from set $\{10^{-i}\}_{i=1}^4$ and random variable χ was generated from $0.6\cdot U(0,1)$ in each generation.

In the second part, we compared the proposed MFEA-DGD with the existing MTES algorithm [54], which also provides a theoretical analysis of the convergence and adopts OpenAI ES to simulate gradients. Note that the MTES algorithm does not have open source code available; thus, we follow the experimental parameters and results presented in the corresponding literature [54]. For the proposed MFEA-DGD, the maximum number of calculations to be changed to $250000\times n$, and the independent number of runs in the experiment was set to 30. The other parameter settings were the same as described for the first part of the experiments.

C. Results and discussion

In the following, we present the experimental results and answer questions Q1, Q2, and Q3.

Q1: First, we consider Q1, and then we compare the proposed MFEA-DGD with five state-of-the-art multitasking methods. The 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 given in Tables V and VI, respectively. These results are used to compare the six algorithms according to Friedman's test at a significance 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 Tables V and VI. Here, the symbols "$-$", "$\approx$" and "$+$" imply that the corresponding method was significantly worse, similar to, and better than the proposed MFEA-DGD on the Wilcoxon rank-sum test with a 95% confidence level, respectively. The best performance values are shown in bold.

As shown in Tables V and VI, the proposed MFEA-DGD performed exceptionally well on the two benchmarks of continuous MTO problems in terms of the averaged objective value. With MFEA, negative transfer is unavoidable because the knowledge transfer occurs randomly. However, the proposed MFEA-DGD simulates the dynamics of the DGD algorithm, which can combine the convexity of different tasks to enhance positive transfer. In some sense, a sufficiently strong positive transfer can offset the harm caused by negative transfer. MFEA-II optimizes the probability of knowledge transfer between tasks; however, its ability to overcome negative transfer is still inferior compared to the proposed MFEA-DGD. In terms of the averaged objective value on test suites 1 and 2, the proposed method outperformed or matched MFEA-AKT on 16 of 18 tasks and 15 of 20 tasks, respectively. This demonstrates that, while MFEA-AKT can select the appropriate crossover operator from SBX, arithmetical, geometrical, and BLX-α adaptively, the new crossover and mutation operators designed in the proposed MFEA-DGD help realize a more effective search for the global optima. The MFEA-GHS also employs the hyper-rectangle search strategy and utilizes

TABLE V: Averaged standard objective value of six compared methods, over 20 independent runs on the single-objective MTO test suite 1. (The *meanrank* was 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 VI: Averaged standard objective value of six compared methods, over 20 independent runs on the single-objective MTO test suite 2. (The *meanrank* was 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±7.56E-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

a general algorithm framework that is the same as that of the proposed MFEA-DGD. A comparison of the results obtained by MFEA-GHS and the proposed MFEA-DGD shows that the proposed operators outperform the SBX crossover, PM, and genetic transform strategy in MFEA-GHS. This reconfirms the benefit of introducing quasi-GD mutation. In addition, MTEA-AD outperforms many existing EMTO methods, and its framework differs from several MFEAs. However, we found that the proposed MFEA-DGD outperformed MTEA-AD on both test suites, which indicates that the proposed algorithm is very competitive.

Q2: Here, we explain the experimental results in terms of the theoretical properties of the DGD algorithm implied by Theorem 1, and we show the average convergence traces of the compared algorithms on all problems in test suites 1 and 2. Due to space limitations, the corresponding figures are provided online at http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/trends.pdf. In these figures, the x-axis represents the number of FE, and the y-axis indicates the average objective value on a log scale. To prevent illegal values on the log scale, the average objective value of a solved task was set to 1E-07. As shown in these results, the proposed MFEA-DGD obtained the fastest convergence rate for most tasks, particularly in the early stage where its convergence rate is remarkable. Thus, the advantage of MFEA-DGD is greater than that of algorithms in cases of insufficient computational resources, which aligns with the geometric convergence rate of the DGD algorithm stated in Theorem 1. Since we have introduced the quasi-GD concept, the convergence rate of the proposed MFEA-DGD is improved compared to other EAs when the local convexity of the tasks is good.

Next, we attempt to elucidate why the proposed MFEA-DGD achieved excellent performance on some problems in test suite 1 while being less effective or even disadvantageous on

TABLE VII: 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	$\approx 0.5^d$	$B \to \infty$

other problems by combining the properties of the functions in Table III and Theorem 1. In light of Theorem 1, it follows that for each pair of twin tasks, the DGD algorithm can converge quickly if their global optima are close and satisfy the joint strong convexity in Assumption 2 (Assumption 1 obviously holds in the test suite 1 setting, and Assumption 3 can be guaranteed by selecting appropriate hyperparameters). Thus, 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, i.e., the distance between the global optima of the two tasks and the strong convexity of the sum of the twin tasks. The former can be observed directly from Table III, where the last column shows the Euclidean distance between the global minima of each pair of twin tasks on the unified expression space $[-1, 1]^{50}$. For the latter, we used the following quantity to measure the strong convexity of the function. Here, 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 of dimension d. Clearly, the size of $C(f)$ measures the strong convexity of the function f. Based on this definition, we briefly review the strong convexity of the classes of the functions used in test suite 1 [68]. After simple estimations and calculations (additional 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 VII with $\Theta = [-B, B]^d$, which allows us to compare the convexity of different pairs of twin tasks in test suite 1. Note that $C(Schwefel) \approx 0.5^d$ in Table VII implies that $\lim_{B\to\infty} C(Schwefel) = 0.5^d$; however, we cannot determine whether $C(Schwefel) - 0.5^d$ 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 obtain their convexity ranking in the general search space. Here, we must verify the above analysis by combining the experiments results.

According to the level of distance of global minimums, we divide the nine problems in test suite 1 into four subsets to facilitate this discussion:, $\{F_1, F_2, F_3, F_6\}$, $\{F_5, F_7\}$, $\{F_4, F_8\}$, and $\{F_9\}$. First, for problem F_1,F_2,F_3, and 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). Using the concept of control variables, we only need to focus on the convexity of each pair of twin tasks. Since $C(Griewank) + C(Rastrigin) > C(Ackley) + C(Rastrigin) > C(Ackley) + C(Weierstrass)$, the order of the convexity ranking of F_1, F_2, and F_6 is exactly the same as MFEA-DGD's ranking of the best fitness above F_1, F_2, and F_6 from smallest to largest (Table V). Similarly, by $C(Griewank) + C(Rastrigin) > C(Ackley) + C(Schwefel)$, this is consistent with the comparison of the average objective values of F_1 and F_3 in Table V. Next, we consider problems F_5 and F_7. The distances between their global minimums are 0.1 and 0.1414, respectively, which are similar but not precisely equal values. Note that for $C(Rosenbrock) + C(Rastrigin) > C(Ackley) + C(Rosenbrock)$, 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 problems F_4 and F_8 where the global minimums are more distant, with $C(Rastrigin) + C(Sphere) > C(Griewank) + C(Weierstrass)$. Thus, the convexity of F_4 is better than that of F_8, and according to our theory, the proposed MFEA-DGD may perform better on problem F_4, which is consistent with the experimental results. Finally, it is easy to find that the proposed MFEA-DGD performs the worst on problem F_9, and the reason can be well explained because the distance between the global minimums of the twin tasks in problem F_9 is much greater than that of the other problems; thus, theoretically, the proposed MFEA-DGD algorithm will not perform well on such problems.

Q3: The MTES algorithm was the first multitask EA in existing literature that simultaneously establishes a strict convergence analysis with the GD approximation. Compared to MTES, the main advantage of the proposed MFEA-DGD is the combination of the MFEA framework and GD, which makes the algorithm much more scalable and enhances the population diversity. In addition, our convergence analysis does not assume the convexity of each task, which is realistic. Table VIII compares the performance of the two algorithms on test suite 1. As can be seen, the proposed MFEA-DGD significantly outperformed MTES on nearly all tasks with excellent performance.

To investigate the effects of the key components of the proposed MFEA-DGD, we compared it to two variants, i.e., (1) replacing the crossover operator in MFEA-DGD with SBX and (2) replacing the mutation operator with PM. The results showed that the GD introduced in the proposed MFEA-DGD plays a fundamental role in improving the algorithm's performance. Due to the page limitations, the detailed experimental results are provided in Table I and Figs. 11-15 in the online document (http://csse.szu.edu.cn/staff/zhuzx/MFEA-DGD/trends.pdf).

VI. Conclusions

In this paper, we have proposed a method to handle multitasking with nonconvex tasks. We have also provided theoretical proof that the DGD method effectively handles these tasks while exhibiting rapid convergence. To further distinguish

TABLE VIII: Objective function values of test suite 1 obtained by MFEA-DGD and MTES algorithms (the results are averaged over 30 runs).

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

our work, we have introduced the MFEA-DGD algorithm, which represents a groundbreaking expansion of the MFEA method. The proposed MFEA-DGD algorithm incorporates new reproduction operators and an effective hyper-rectangle search strategy. The proposed MFEA-DGD is characterized by innovative crossover and mutation operators that simulate the dynamics of the DGD algorithm. We also employ OpenAI ES to estimate the unknown gradient. The interpretability of the role of the crossover and mutation operators in the proposed MFEA-DGD is derived from DGD's convergence analysis, thereby providing a distinct advantage over existing methods. Specifically, the crossover operator combines the local convexity between similar tasks, and the mutation operator leverages GD to identify superior offspring. In addition, the hyper-rectangle strategy expands the algorithm's search range. Our contributions were validated by comparing the proposed MFEA-DGD to existing EMTO algorithms on two MTO test suites. The experimental results demonstrate the superior performance of the proposed method. We have also offered a valid theoretical explanation for the experimental results by introducing a measure of convexity for different tasks.

Despite the promising performance of the proposed MFEA-DGD, there is room for improvement. For example, the performance of MFEA-DGD on MTO problems containing greater than two tasks should be investigated further, and the application of MFEA-DGD to real-world problems should be investigated extensively. Furthermore, the proposed MFEA-DGD introduces the quasi-GD; thus, it may also have great potential to solve systems of nonlinear equations [69]. The idea of incorporating quasi-GD into EAs can also be applied to solve multi-objective optimization problems [70]; thus, the proposed MFEA-DGD method could be extended to multi-objective optimization problems.

The source code of MFEA-DGD written in MATLAB is available online 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," in *Artificial Life and Computational Intelligence: Second Australasian Conference, ACALCI 2016, Canberra, ACT, Australia, February 2-5, 2016, Proceedings 2*. Springer, 2016, pp. 54–65.

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

[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," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–8.

[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)*, vol. 26, no. 5, pp. 941–960, 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," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 2404–2411.

[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] X. Wu, W. Wang, H. Yang, H. Han, and J. Qiao, "Diversified knowledge transfer strategy for multitasking particle swarm optimization," *IEEE Transactions on Cybernetics (Early Access)*, pp. 1–14, 2023.

[33] X. Wang, Q. Kang, M. Zhou, S. Yao, and A. Abusorrah, "Domain adaptation multitask optimization," *IEEE Transactions on Cybernetics (Early Access)*, pp. 1–12, 2022.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] S.-H. Wu, Z.-H. Zhan, K. C. Tan, and J. Zhang, "Transferable adaptive differential evolution for many-task optimization," *IEEE Transactions on Cybernetics (Early Access)*, pp. 1–14, 2023.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] Y. Xu, C. Xu, H. Zhang, L. Huang, Y. Liu, Y. Nojima, and X. Zeng, "A multi-population multi-objective evolutionary algorithm based on the contribution of decision variables to objectives for large-scale multi/many-objective optimization," *IEEE Transactions on Cybernetics*, 2022.

[53] D. Li, L. Wang, L. Li, W. Guo, Q. Wu, and A. Lerch, "A large-scale multiobjective particle swarm optimizer with enhanced balance of convergence and diversity," *IEEE Transactions on Cybernetics*, 2022.

[54] 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.

[55] 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.

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

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

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

[65] 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.

[66] 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.

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

[68] 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.

[69] A. Song, G. Wu, W. Pedrycz, and L. Wang, "Integrating variable reduction strategy with evolutionary algorithms for solving nonlinear equations systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 75–89, 2021.

[70] Y. Tian, H. Chen, H. Ma, X. Zhang, K. C. Tan, and Y. Jin, "Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 10, pp. 1801–1817, 2022.