Multipopulation Optimization for Multitask Optimization

Zedong Tang, Maoguo Gong*, Fenlong Jiang, Hao Li
The School of Electronic Engineering
the Key Laboratory of Intelligent Perception and
Image Understanding of Ministry of Education
Xidian University
Xi'an, Shaanxi Province 710071, China
*Email:gong@ieee.org

Yue Wu
The School of Computer Science and Technology
Xidian University
Xi'an, Shaanxi Province 710071, China
Email:ywu@xidian.edu.cn

***Abstract*—Currently, the most of multitask evolutionary algorithms views multiple tasks as factors influencing the evolution of individuals. However, this consideration causes difficulty to assign fitness to individuals, because an individual which performs well on one task can have a bad performance on another task. To avoid this difficulty, this paper proposes a novel multipopulation technique for multitask optimization (MPMTO). The novelty of MPMTO is that it can solve the multiple tasks via a simple and straightforward method by corresponding each population to a task. By this way, the fitness assignment issue can be addressed by just assigning the objective value of the corresponding task to individuals. MPMTO is a general technique so that existing population-based optimization algorithms can be used in each population. This paper uses differential evolutionary algorithm in each population and develops a multipopulation multitask differential evolutionary optimization (mMTDE) based on the proposed multipopulation technique. mMTDE features that each population can use the other populations as the additional knowledge source to create an overlapping population, allowing the populations share information. By this way, the population can improve the efficacy and accuracy of solving multiple tasks. Moreover, the successful inter-task offspring can immigrate back to the corresponding population to fully utilize the inter-task knowledge. We have compared the proposed method with other state-of-the-art methods on benchmark multitask problems. The experimental results show the superiority of the proposed method which could utilizes efficiently the searching knowledge of multiple tasks.**

I. Introduction

Multitask optimization (MTO) [1] is a newly emerging research theme in the field of evolutionary optimization algorithms (EAs), which utilizes the underlying similarity of optimization tasks and facilitates the transfer of the inter-task knowledge to efficiently solve multiple heterogeneous tasks. Two concerns share many commons with the multitask learning (MTL) [2] [3] in the field of machine learning, which has the similar target on using the additional information sources to improve the performance of learning or searching.

An efficient way to utilize these sharing knowledge is to adopt a population-based searching algorithm, like genetic algorithm (GA), differential evolution (DE) and particle swarm optimization (PSO), and incorporate an information transfer strategy into it, leading to evolutionary multitasking [4]. Currently, there exist some works focusing explicit reusing of the knowledge from the similar problems to solve a similar but more complicated incoming problems [5]–[9]. Multifactorial optimization (MFO) [4] takes several diverse tasks in account in a single population. The term "factor" is referred as the component task in the framework of MFO, which influences the evolution of individuals. There is no need to have the relationship of tasks *a priori*, because a MFO algorithm can handle the transfer of the shared information implicitly. There are two benefits of MFO against conventional batch optimization, a) Improvement of the accuracies of component tasks because of knowledge transfer and reutilization; b) Reduction of the total make-span achieved by reusing the existing knowledge in the similar tasks and not searching from scratch. Currently, a specified MFO algorithm integrated into GA, called multifactorial evolutionary algorithm (MFEA), was proposed in [4]. MFEA is the first instructive work in the unexplored area of evolutionary multitasking. MFEA, inspired by the concept of multifactorial inheritance, is characterized by the presence of the cultural traits and a subtle offspring reproducing strategy. By utilizing the concept of multifactorial inheritance, MFEA improves the searching ability of GA via the cross-domain transfer of information and achieves a considerable accelerating convergence speed and utilize the implicit parallelism of EAs. MFEA has already been utilized to solve single-objective optimization (SOO) [4] and multiobjective optimization (MOO) tasks [10] and achieved impressive results.

The researchers developed many works that the heterogeneous populations are used to solve the complex problems in multiobjective and large-scale problems. The coevolutionary algorithms with the multipopulation technique can ameliorate the difficulty of fitness assignment and computational efficacy. Therefore, in this paper, we propose a novel multipopulation technique for solving multitask optimization (MPMTO). The proposed multipopulation technique is a general mechanism. Existing evolutionary algorithms can be used in each population. Due to the

978-1-7281-2153-6/19/$31.00 ©2019 IEEE

simplicity and computational efficiency, differential evolutionary algorithms (DEs) are widely employed for coping with many real-world problems. Enlightened from the superiority of DEs, in this paper, we employ an improved differential evolutionary algorithm in the proposed multipopulation technique, developing a multipopulation multitask differential evolutionary algorithm (mMTDE). We show our contributions as follows.

- A light-weight multipopulation framework is developed to tackle with the fitness assignment problem in the multitask optimization. Populations can share the search knowledge by creating overlapping populations.
- The successful individuals generated by inter-task crossover can replace the inferior individuals for promoting the propagation of the inter-task search information.
- To balance the exploration and the exploitation, we propose an improved differential evolutionary algorithm, which utilizes mutation strategies with different search ranges at different stages of the evolution process.
- To address the difficulty of parameter settings, we employ an adaptive strategy to adjust the relative parameters.
- We validate our method on comprehensive benchmark problems which consist of two or three tasks. The experimental results show the superiority of our method.

The rest of this paper is organized as follow: Section II will describe related background, including the definition of MFO and the existing works on evolutionary multitasking. The proposed multipopulation multitask differential evolutionary algorithm (mMTDE) is described in Section III. In Section IV, the performance of the proposed algorithm is also evaluated compared with the state-of-the-art MTO methods in Section IV. In Section V, the work in this paper is concluded.

II. BACKGROUND

A. Multitask Optimization

Multitask optimization is an emerging research field aiming at utilizing the transfer of searching experience of solving a task to assist solving of another task. It is worth to note that multitask optimization is a composite problem which considers the optimization tasks simultaneously that have traditionally been regarded as distinct tasks. MTO attempts to fully and simultaneously optimize each task which may have distinct search spaces and different optima. Consider a composite problem consisting of K tasks. The problem of MTO is formulated explicitly as follows: $\{\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_K^*\} = \mathrm{argmin}\{f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), ..., f_K(\mathbf{x}_K)\}$. A canonical MTO framework, called evolutionary multitasking, has been put forward in [10], where the nature-inspired population-based methods are leveraged to solve MTO problems by unleashing the potential of the implicit parallelism featured by EAs.

MFO is a popular evolutionary multitasking paradigm currently, since evolutionary multitasking has been proposed. In particular, each f_i among the composite problem can be viewed as an existing factor influencing the evolution of individuals in the K-factorial environment [11]. In multifactorial optimization, the specific tasks' search spaces are mapped into a unified representation space. The MFO algorithms use a single population to explore and exploit the unified representation with a unified comparing criterion. MFO features assortation mating, selective imitation and selective evaluation. Assortation mating is a key characteristic which enables the implicit transfer of knowledge of solving tasks by utilizing the crossover operators (or other nature-inspired operators). Selective imitation enables the transfer of cultural traits, which promotes the propagation of good structures of solutions. For the sake of reducing the computational cost, the selective evaluation only evaluates each individual on one of tasks associated to it under the principle.

MFEA is the first implementation of MFO which corporates GAs into the MFO paradigm [4]. MFO uses an elaborated performance criterion to evaluate individuals. The factorial rank is used to represent the relative performance of p_i on jth task. Therefore, the fitness function of MFO is defined as the inverse of an individual's minimal factorial rank over all tasks. This fitness function is viewed as a unified performance index in the multifactorial environment. The evolution mechanism is incorporated with multifactorial inheritance, assortative mating and vertical cultural transmission, to deal with the multiple environments in MFO. Assortative mating is an important principle which implements the cross-domain transfer of genetic materials. Vertical cultural transmission implements the behavior of offspring imitating from parents across different cultural pedigree via selective imitation strategy.

There are some works involving variants of MFEA and applications on real-work problems. Yuan *et al.* [9] proposed a variant of MFEA for permutation-based combinatorial optimization problems (PCOPs). This work contributes a new unified representation scheme for the PCOP and a level-based selection procedure to improve the performance on PCOPs. Bali *et al.* [12] proposed to replace the unified representation scheme with a linear domain adaption strategy. The linear domain adaption strategy (LDA) computes a linear map from the decision space of one task to another directly, and the inverse map takes the reverse process. This method achieves significant performance improvement on the benchmark problems as reported in [12]. Sagarna *et al.* [13] applied MFEA on the software tests generation. This work focuses on branch searching and is the first time to apply MFEA to real-world problems with more than two tasks. Gupta *et al.* [10] proposed a variant MFEA for multitasking multi-objective which shows its benefit on optimizing the rigid-tool Liquid Composite Molding (LCM) processes. A realization of multifactorial multi-objective optimization was presented in this work. Currently, Li *et al.* [14] proposed a multipopulation frame for multifactorial optimization. As this work shares some commons with our method, a comparison in detailed are conducted in the end of Section III.

B. Differential evolution

Differential Evolution (DE) [15] [16] [17] is a global optimization algorithm, which starts with a set of well spreading positions generated randomly with the uniform distribution in the range [0,1]. These randomly generated positions form a population $\{X_{i,G}, i = 1, ..., N_p\}$, called target vectors. After initializing the target vectors, DE uses the mutation operation to make a trail vector according the target vectors. There are several mutation strategies frequently used in the literature, such as DE/rand/1, DE/current-to-best/1, DE/best/1 etc. Then, trail vectors are combined with target vectors to generate the next generation by the crossover operation. For the *i*th individual, we produce a list of randomly real numbers $r_1, r_2, ..., r_d, ..., r_D$ as large as the target vector. Then, the element in each dimension of the individual for the next generation is determined by below equation,

$$x_{i,G}^d = \begin{cases} u_{i,G}^d & \text{if } r_d < CR \text{ or } d = R_i, \\ x_{i,G-1}^d & \text{else.} \end{cases} \tag{1}$$

where *CR* is the user-specified parameter controlling the crossover probability. r_i is a random number in the range [0, 1]. If $r_i < CR$, the element in the associated dimension copies the value from the trail vector respectively. Otherwise, the element copies the value from the target vector. R_i is a random integer number in the range $[1, D]$. R_i is a random integer in the range $[1, D]$, used to guarantee the generated vector always has at least one dimension different from the target vector. The application of DE and its variants in and realistic problems indicates its effectiveness and practicability. In this paper, we choose DE as the kernel optimization algorithm of the proposed methods.

III. Methodology

In this section, a multi-population framework for multitasking optimization (MPMTO) is first elucidated. Afterwards, the proposed framework is incorporated into differential evolutionary algorithm to develop a multipopulation multitask differential evolutionary algorithm (mMTDE). In order to isolate the information of each task, the proposed algorithm creates multiple populations each of which corresponds to a task. A novel inter-task knowledge transfer technique, based on the individual immigration, is employed in the proposed multipopulation multitask algorithm to utilize the additional source of search information, thus, enhances the performance of the algorithm on each task. In addition, when the offspring that is generated by the combination of individuals from different tasks survives in the next generation, these offspring can benefit the tasks where the parents come. Therefore, at the end of each iteration, these surviving offspring replace the inferior individuals of the aforementioned task. For example, assume that there are three tasks to be solved. The relationship of tasks could be depicted as a topology where each vertex represents a task and the bio-direction edge represents the shared knowledge transfer. As depicted in Fig. 1, T_1 has two edge which connects from it to other tasks. Therefore, the search knowledge can propagate from two tasks to T_1. We employ the differential evolutionary algorithm as the core optimization algorithm of the proposed multipopulation technique to solve multitask optimization problems. In addition, the parameters in DE significantly influence the performance algorithm. Thus, the values of parameters are adaptive tuned in the process of evolution.

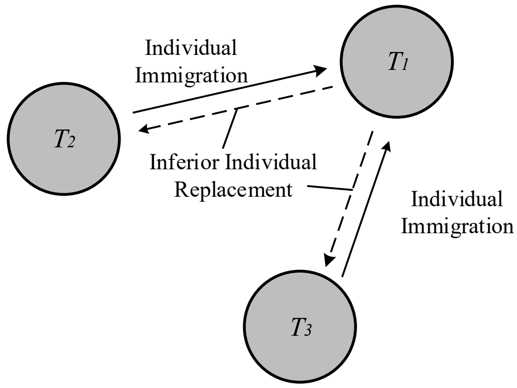

Fig. 1. The overview of the multipopulation technique for multitask optimization.

A. Overlapping Population Creation

In Algorithm 3, the main structure is similar to the canonical differential evolutionary algorithm, but the multipopulation framework and the transfer mechanism of problems-solving knowledge are incorporated into the algorithm. The subpopulation creation process executes every ΔG generations to transfer the inter-task search knowledge and increase the diversity of each subpopulation. The differential evolutionary algorithm is used in each subpopulation. The DE operators will generate a subpopulation of offspring each iteration, then the overall population is updated by replacing the corresponding individuals with the elites of the subpopulation.

The proposed MPMTO framework divides the population into several populations, and each of them corresponds to one task. By this way, each population can be evaluated straightforward by utilizing the objective function value. Each subpopulation focuses on solving one task. The implicit transfer takes place inter-subpopulation. In the

Algorithm 1 Overlapping Population Creation Procedure

Input: Overall Population P, Objective Value F_t of Each Task T_t, the number of elite individuals N_t and the number of transferred individuals N_μ.

Output: Subpopulation P_t of Task T_t.

1. Sort the individuals by ascend order of objective values.
2. Select the top N_t elites into the subpopulation P_t.
3. Shuffle the rest of individuals.
4. Select N_μ individuals from them and append these individuals into P_t.

original multifactorial optimization, the selection strategy, that two parents from different tasks are chose to generation offspring, is employed to implement the implicit transfer. However, in the proposed dynamic resource allocation strategy, the subpopulation is regarded as a chunk which the algorithm authorizes continuing the evolving process. Therefore, the subpopulation is constructed by a subset of the overall population. First, the composite subpopulation N_t' includes all individuals belonging to T_t and the extra individuals selected from other tasks. The number of extra individuals is determined by the learning parameter μ. Then canonical differential evolutionary operations perform on the composite subpopulation including differential mutation, crossover and selection.

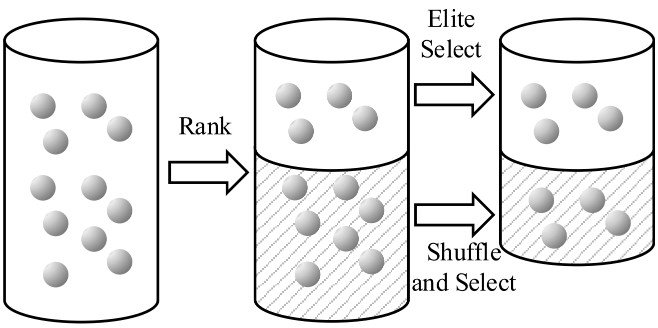

Fig. 2. The process of overlapping population creation for each task from the overall population.

Because a composite subpopulation replaces the original population before solving a specific task, a subset of the overall population is selected by an altered strategy. The composite subpopulation consists of an extra number of individuals and the main individuals assigned to a task. The number of transferred individuals, denoted as N_μ, is the product of learning parameter μ and the number of individuals N_t explicitly assigned to task T_t as shown below,

$$N_\mu = max\{\lfloor \mu \times N_t \rfloor, N_{\mu,min}\}. \tag{2}$$

where N_μ is the number of the injected individuals, N_t is the individuals assigned to T_t and μ is the learning parameter which controls the agree of the cross-domain implicit transfer. In additional, the $N_{\mu,min}$ is the minimal number of injected individuals which guarantees the diversity of subpopulation and reduces the speed of convergence. Next, N_μ individuals is selected through the tournament strategy according to the factorial ranks of them. And the subpopulation totally includes $N_\mu + N_t$ individuals. Once the subpopulations are established, MPMFO treats the individuals in the same subpopulation equally. Thus, a canonical EA can be used to evolve each population.

B. Improved multitask framework incorporated into DE

The proposed multipopulation technique is a general technique, where the populations can use the existing evolutionary optimization. In this paper, we adapt differential evolutionary algorithms (DEs). After populations corresponding to every task are built. Each population can use the DE mutation and crossover to generate the next generation. The process of evolution would keep running until reaching the terminal condition.

We use two differential mutation operations in the proposed algorithm, which have different search ranges. The first mutation operation is the DE/rand/1/bin operation. This operation has a relatively large search range which is used to promote the exploration of a broad area. The trail vectors are generated by combination of the three individuals selected from the population randomly as shown in the following equation.

$$u_{i,G}^d = x_{r1(i),G}^d + F \cdot (x_{r2(i),G}^d - x_{r3(i),G}^d)) \tag{3}$$

where G means the current generation, and $r1$ and $r2$ and $r3$ are the lists of random integer numbers in the range [1, N_j], N_j indicating the number of individuals in the jth population. The second mutation operation employed in this paper is the DE/pbest/1/bin. The elite information is incorporated into the generation of the offspring. Therefore, we use this relatively exploitive mutation operation to improve the refine search at the later stage of the evolution. The corresponding equation is shown as follow.

$$\begin{aligned} u_{i,G}^d = x_{i,G}^d &+ F \cdot (pbest - x_{i,G}^d) \\ &+ F \cdot (x_{r2(i),G}^d - x_{r1(i),G}^d) \end{aligned} \tag{4}$$

where the vector $pbest$ of each individual is selected randomly from the top-p elite individuals. We set $p = 5\%$ recommended in [18]. The algorithm uses the first mutation operation at the beginning of the evolution process. After a half of the maximum function evaluation calls, the second mutation operation is activated.

After that, the trail vectors are combined with the current solutions to generate the offspring.

$$x_{i,G}^d = \begin{cases} u_{i,G}^d & \text{if } r_d < CR_i \text{ or } d = R_i, \\ x_{i,G-1}^d & \text{else.} \end{cases} \tag{5}$$

When using the crossover operation to generate the trail vector, some good structure in the population might be destroyed which is unexpected to get good performance. To protect the good structure of current good position, the crossover probabilities of each individual are different each other. If an individual is fitter, the crossover probability is relatively small. So, the crossover probability is determined by using the following equation:

$$CR_i = 1.1 - exp(-\theta_i/N_j). \tag{6}$$

where CR_i ranges round 0.1 to 0.7, θ_i is the index of p_i determined by the ranking of individuals. N_j is the subpopulation size belonging to T_j. The crossover operation is normally applied on individuals by equation (5). As recommended in [18], we sample the F_i for each individual from a Cauchy distribution with the scale parameter of 0.1.

The location parameter of the distribution is calculated by using the Lehmer mean of the successful F during each iteration.

It is worthwhile to note that the parents of an offspring can come from different task, since we build the subpopulation for the corresponding task by immigrating the individuals from multiple tasks. By this way, though we use a canonical form of DE operations herein, the algorithm can use the knowledge of multiple tasks to generate a population with high diversity. The transfer of these inter-task information can improve the convergence speed and accuracy on the problems sharing underlying similarity.

C. Immigration of Successful Individuals

An offspring, generated from the inter-task crossover and surviving in the next generation, is called successful individuals. If two tasks are complementary, the individual generated via the inter-task crossover might have a large probability to be a good solution for both tasks. So the inverted transfer of individuals can facilitate the propagation of the building blocks generating in the search process. However, individuals from two similar tasks can reproduce a bad solution probabilistically. For this reason, in the proposed method, we only consider replacing the inferior individuals by the surviving individuals.

In the subpopulation creation process, the task's index, from which the individuals are selected, are recorded in a list L_j, j indicating the jth population corresponding. When generating the offspring, the algorithm restores the random numbers $r2$ used in equations (3) and (4). At the end of each iteration, we check each surviving individual. As long as a superior individual is generated with the immigrated individual guiding, it is expected that these two populations can both benefit from the transfer of information. Therefore, conducting the inverted transfer process is reasonable in the term of taking advantages of genetic materials. In the other words, if $L_j(r2(i))$ is not equal to $L_j(i)$, then the interior individual in the task $L_j(r2(i))$ is replaced by the corresponding individual.

Algorithm 2 Successful Individual Immigration Procedure

Input: Subpopulation P_j of current task, Set of Subpopulations of other tasks $Q = \{P_t | t \in T \backslash \{j\}\}$.

```
for each x_i ∈ P_j do
    k = L_j(r2(i))
    if k ≠ L_j(i) then
        Sort the individuals in P_k ∈ Q.
        Replace an inferior individual within the last p% individuals
        by x_i. #p is set to 5.
    end if
end for
```

D. Algorithm

In the proposed method, called multipopulation multitask framework, each task has an isolated population and share the searching experience by employing an individual immigration strategy. The population is first split into K subpopulation associated with K tasks. Then, a number of individuals from other tasks can immigrate to the subpopulation. By this way, one individual can use the information from multiple tasks instead of a single task, as the immigrated individuals are selected from multiple populations.

The differential evolutionary algorithm is employed herein to evolve every subpopulation, which develops a multipopulation multitask differential evolutionary algorithm. The main structure of the multipopulation multitask differential evolutionary approach is shown in **Algorithm 3**. At the beginning of each iteration, the populations corresponding to every task are build up. We construct these populations by employing an individual immigration technique, which can combine the inter-task information by transferring the individuals from multiple tasks. The details of the subpopulation creation process are listed in **Algorithm 1**. Then in lines 16-22 in **Algorithm 3**, the DE mutation and crossover are sequentially applied on the population to generate the next generation of the population. In line 29, the individuals in the overall are updated by replacing the individuals corresponding to the current task with the superior ones in the subpopulation. At the end of each iteration, we select the surviving individuals which are generate by the inter-task crossover in the new generation, and replace the inferior individuals of the corresponding tasks by these individuals in line 30. The immigration process is listed in details in **Algorithm 2**.

E. MPMFO versus MPEF-SHADE

MPEF-SHADE is another similarity technique proposed in [14], where multiple populations are employed to solve multiple tasks. However, there are several differences between the proposed MPMFO and MPEF-SHADE. The distinctions are summarized in details as follows,

1) The MPMFO exploits the inter-task information via the immigration of the highly potential individuals. In contrary, MPEF-SHADE employs the DE mutation to exploit the inter-task information. The individuals particularly is guided by a best individual of randomly selected tasks.
2) In MPMFO, the successful individuals can be re-utilized, which is profitable to exploit the shared knowledge in the population. MPEF-SHADE lacks of such mechanism.
3) Once the subpopulations are established , MPMFO treats the individuals within the same subpopulation equally and evolves by a consistent evolutionary mechanism. By this means, MPMFO can keep the similarity and computational efficiency consistently.
4) An improved differential evolutionary is devised to aid MPMFO by compromising the inter-task exploration and intra-task exploitation.

IV. Experiments

In this section, the performance of proposed mMTDE is evaluated compared with other two MFO algorithms and one

Algorithm 3 Basic Structure of mMTDE

Input: The population size N_P, the number of tasks K, the set of tasks $\{T_1, T_2, ..., T_K\}$, the maximum generation G_{max}.

Output: The population at the final stage P^*.

Initialize a population of N_P individuals in P.
Evaluate the objective values corresponding each individual of P.
Compute the skill factor τ_i of each individual p_i.
Let $N_t^* = \lfloor N_P/K \rfloor$.
Let $g = 1$.
repeat
  Let $P_t = P_t \cup P_\mu$.
  Compute the size of external task group N_μ by equation (2).
  Let $N_t = N_t + N_\mu$.
  Construct the subpopulation for T_t via **Algorithm 1**.
  Record the task index of each individual in P_t, denoted as L_t.
  for each T_t in $\{T_1, T_2, ..., T_K\}$ **do**
    Let $G = 1$.
    while $G \leq \Delta G$ **do**
      for $i = 1$ to N_t **do**
        Calculate F_i and CR_i as described in Section III-B.
        Generate the exclusive random integer numbers r_1, r_2, r_3. # mutation
        if $g < 0.5 \times G_{max}$ **then**
          Generate the trial vector by equation 3.
        else
          Generate the trial vector by equation 4.
        end if
        Let $\mathbf{x}_{i,G} = crossover[\mathbf{x}_{r_1,G-1}, \mathbf{u}_{i,G}, CR_i]$.
        Evaluate $\mathbf{x}_{i,G}$.
        Let $C_G = C_G \cup \{\mathbf{x}_{i,G}\}$.
      end for
      Let $P_t = select[P_t \cup C_G]$.
      Let $G = G + 1$.
    end while
    Select the top N_t individuals in P_t and replace the individuals corresponding to T_t in the overall P.
    Replace the inferior individuals of the other population by the survivors via **Algorithm 2**.
    Update the mean of F based on the successful Fs.
  end for
  Let $g = g + K * \Delta G$.
until $g \geq G_{max}$
Let $P^* = P$.

SOO algorithm. First, the efficient of the proposed dynamic resource allocation strategy is evaluated. In this part of experiment, mMTDE will be compared with single-objective differential optimization (DE) and two MFO methods, MFEA and MFDE, on a comprehensive suite of multitask problems. These composite problems are composed of independent optimization tasks with different complexity and dimensionality, in additional to composite problems with differential properties.

In order to compare the performances of different algorithms for single and multitasking single-objective optimization, we run the three algorithms on the benchmark set and measure a popular metric for the multitasking benchmark sets in 30 trials before reaching the maximum number of evaluation calls. First, we introduce the metrics using in this part of experiment. The mean and standard deviation of best objective function values are the commonly using metrics to measure the advantages of algorithms' global searching and stability. Further, in the part of comparison, the minimum values of obtained optimal objective function values are also record to present the performance of algorithms in worst cases and best cases.

A. Benchmark Functions

In order to valid mMTDE in terms of convergence speed and quality of solutions, we choose a set of benchmark problems including 9 two-task composite problems.

TABLE I
CEC'2017 TWO-TASK BENCHMARK PROBLEMS 1 – 9

Problem Sets	Task		
	Function	Variants Ranges	Optimum
Set 1	Griewank	$[-100, 100]^{50}$	${}_{50D}(0, ..., 0)$
	Rastrigin	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
Set 2	Ackley	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
	Rastrigin	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
Set 3	Ackley	$[-50, 50]^{50}$	${}_{30D}(42.0969, ..., 42.0969)$
	Schwefel	$[-500, 500]^{50}$	${}_{50D}(420.9687, ..., 420.9687)$
Set 4	Rastrigin	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
	Sphere	$[-100, 100]^{50}$	${}_{50D}(0, ..., 0, 20, ..., 20)$
Set 5	Ackley	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0, 1, ..., 1)$
	Rosenbrock	$[-50, 50]^{50}$	${}_{50D}(1, ..., 1)$
Set 6	Ackley	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
	Weierstrass	$[-50, 50]^{25}$	${}_{25D}(0, ..., 0)$
Set 7	Rosenbrock	$[-50, 50]^{50}$	${}_{50D}(1, ..., 1)$
	Rastrigin	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
Set 8	Griewank	$[-100, 100]^{50}$	${}_{50D}(10, ..., 10)$
	Weierstrass	$[-0.5, 0.5]^{25}$	${}_{25D}(0, ..., 0)$
Set 9	Rastrigin	$[-50, 50]^{50}$	${}_{50D}(0, ..., 0)$
	Schwefel	$[-500, 500]^{50}$	${}_{50D}(420.9687, ..., 420.9687)$

In [19], the benchmark problems construct based on the Spearman's rank correlation and the degree of intersection in the unified search space. Totally 9 sets are proposed with different degrees of inter-task synergy and degrees of intersection in unified search space. The details of the benchmark problems are shown in Table I.

TABLE II
DETAILS OF THE THREE-TASK BENCHMARK PROBLEMS.

Problem Sets	Task		
	Objective	Variable Range	Optimum
Set 10	Ackley	$[-50, 50]^{30}$	$(0, ..., 0)^{30}$
	Weierstrass	$[-50, 50]^{30}$	$(0, ..., 0)^{30}$
	Rastrigin	$[-50, 50]^{30}$	$(0, ..., 0)^{30}$
Set 11	Rastrigin	$[-50, 50]^{30}$	$(10, ..., 10, 0, ..., 0)^{30}$
	Weierstrass	$[-50, 50]^{30}$	$(0, ..., 0)^{30}$
	Rosenbrock	$[-50, 50]^{30}$	$(0, ..., 0, 10, ..., 10)^{30}$

B. Parameter Settings

For the fair comparison, we do not use the local search in the experiments. Parameter settings for the proposed mMTDE and other algorithms are listed as follow.

a) mMTDE

- Population size, $N_p = 200$.
- Maximum NFEs, $MAX_{NFE} = 5 \times 10^5$.
- The subpopulation reformation period, $\Delta G = 20$.

- Crossover probability constant, $C_r = 0.9$.
- Scale factor, $F = 0.5$.
- Random mating parameter, $rlp = 0.1$.

b) MFEA

- Population size, $N_p = 200$.
- Maximum NFEs, $MAX_{NFE} = 5 \times 10^5$.
- Mutation probability constant, $M_r = 0.1$ [4].
- Random mating probability, $rmp = 0.3$ [4].

c) MFDE

- Population size, $N_p = 200$.
- Maximum NFEs, $MAX_{NFE} = 5 \times 10^5$.
- Crossover probability constant, $C_r = 0.9$ [20].
- Factor, $F = 0.5$ [20].
- Random mating probability, $rmp = 0.3$ [20].

d) DE

- Population size, $N_p = 200$.
- Maximum NFEs, $MAX_{NFE} = 5 \times 10^5$.
- Crossover probability constant, $C_r = 0.9$ [17].
- Scale factor, $F = 0.5$ [17].

To fairly compare the convergence performance of the algorithms, the local search is disabled in mMTDE and all compared algorithms. In the following experiments, the above parameters stay the same for all experiments unless the changes are emphasized for some experiments. For all conducted experiments, each reported metrics values are average values over 30 independent runs.

C. Comparison on the Benchmark

In order to evaluate the performance of mMTDE, we compare the proposed method with MFEA which is a canonical multitasking evolutionary algorithm. Herein, the algorithms are given enough computational resources (i.e. $EF_{max} = 10^4 \times D$ where D is the dimensions of a problem). The parameter settings are the same as what is mentioned above. And all algorithms repeat 30 times.

As what one has seen in Table III, mMTDE has achieved the-state-of-art results on almost all of 9 benchmark problems in terms of the best function values. Though, in some benchmark problems, the proposed mMTDE does not obtain the best result in terms of the obtained global optimal solution compared with DE does, but obtains the good averaged performances, like in S9. It shows the advantage of mMTDE on handling the inter-task information transfer and the refining search. The mMTDE achieves the best results in the two-task experiments.

D. Three-task problems

In this section, to assess the scalability of the proposed algorithm, we test mMTDE on the three-task problems. The results are shown in Tables IV and V. It is observed in Table IV that the algorithm is competitive over the compared algorithms, MFEA and MFDE, on these problems having three tasks. On the problems Set 10, the mMTDE obtains the best results, in the term of no matter the mean of objective function values or the minimal objective functions. Meanwhile, the optimal solutions of T_2 and T_3 are obtained by using mMTDE. Also mMTDE achieves the competitive results on Set 11, shown in Table V. Set 11 contains the tasks with distinct optimal solutions. The compared MFO algorithms have a poor performance on these problems compared with the results on Set 10. Due to the multipopulation technique employed in the mMTDE, mMTDE can improve the search performance on these problems by isolating the search information of every task. Also the mMTDE can benefit from the search information by the novel transfer strategy.

V. Conclusion

According the framework of MFO, the overall population is implicitly separated into several subpopulations by the skill factors and the problem-solving knowledge is carried by the high-quality individuals. Therefore, we explicitly emphasize the concept of multi-population in the proposed framework for multitasking and utilize a new method to transfer knowledge among tasks as efficient as that in MFO. The multi-population framework for MTO is capable of controlling the cross-domain information transfer. Multiple populations are constructed for multiple tasks respectively. Such every population contains two parts of individuals. Some of individuals are selected by the elite selection with respect to the solved task. In addition, the other individuals are selected from the other populations as the source of the inter-task knowledge. In different stages of evolution process, the proposed algorithm employs different differential mutation operations, which takes the balance of exploration and exploitation. Moreover, to fully utilize the generated offsprings, the successful inter-task offspring surviving in the next generation can immigrate back to the populations from which the parents come. The experimential results on the two-task and three-task benchmark problems show that the proposed multipopulation multitask framework can efficiently and effectively transfer the knowledge of multiple tasks.

In future, we are planning to develop a more efficient way to transfer the shared knowledge which can deal with underlying similarity and distinction, especially in the case of many tasks. Also, a learning strategy based on the statistical information of population are expected to be a good mean of the adaptive setting of additional parameters.

VI. Acknowledge

This work was supported by the National Natural Science Foundation of China (Grant nos. 61772393), the National key research and development program of China (Grant no. 2017YFB0802200), and the Key research and development program of Shaanxi Province (Grant no. 2018ZDXM-GY-045).

References

[1] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: A computer science view of cognitive multitasking," *Cogn. Comput.*, vol. 8, no. 2, pp. 125–142, 2016.

[2] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[3] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

TABLE III

THE MAXIMUM, MINIMUM, MEAN AND STD VALUES OF OBJECTIVE VALUES AFTER MAXIMUM EFS OVER 30 RUNS. THE SUPERIOR PERFORMANCE IS HIGHLIGHTED IN **BOLD**. AND THE SECOND BEST ONE IS TYPED IN *UNDERLINE AND ITALIC* TYPE.

Problem		mMTDE		MFEA		MFDE		DE	
		T_1	T_2	T_1	T_2	T_1	T_2	T_1	T_2
S1	Min	**0.00e+00**	**0.00e+00**	1.18e-02	7.28e+01	*4.00e-03*	*1.39e+01*	9.44e-03	1.89e+01
	Mean	**0.00e+00**	**0.00e+00**	*2.55e-02*	1.49e+02	6.36e-02	*3.52e+01*	5.56e-02	3.91e+01
	Std	0.00e+00	0.00e+00	9.76e-03	4.57e+01	4.28e-02	1.01e+01	4.76e-02	1.10e+01
S2	Min	**8.88e-16**	**0.00e+00**	1.74e+00	5.26e+01	5.90e-01	1.19e+01	*3.17e-01*	*6.97e+00*
	Mean	**1.13e-15**	**0.00e+00**	3.59e+00	1.80e+02	1.48e+00	2.81e+01	*1.27e+00*	*2.21e+01*
	Std	8.86e-16	0.00e+00	6.32e-01	6.33e+01	3.94e-01	7.95e+00	4.87e-01	1.03e+01
S3	Min	**7.99e-15**	**6.36e-04**	3.04e+00	2.01e+03	*4.86e-01*	6.73e-04	2.05e+00	*6.56e-04*
	Mean	**1.36e-05**	**6.36e-04**	1.93e+01	2.57e+03	7.86e+00	1.46e+03	*6.95e+00*	*1.44e+03*
	Std	7.32e-05	0.00e+00	3.03e+00	2.96e+02	2.01e+00	9.56e+02	2.20e+00	1.29e+03
S4	Min	**0.00e+00**	**0.00e+00**	3.68e+02	*2.08e-02*	1.12e+02	2.78e+00	*9.56e+01*	4.75e-01
	Mean	**0.00e+00**	**0.00e+00**	4.81e+02	*3.84e-02*	1.96e+02	2.20e+01	*1.74e+02*	1.45e+01
	Std	0.00e+00	0.00e+00	8.58e+01	1.39e-02	4.16e+01	1.44e+01	4.03e+01	1.22e+01
S5	Min	**3.24e-10**	**4.39e+01**	1.42e+00	9.33e+01	8.26e-04	8.91e+01	*8.23e-04*	*8.17e+01*
	Mean	**7.58e-08**	**7.02e+01**	2.66e+00	1.62e+02	1.63e+00	1.73e+02	*1.62e+00*	*1.61e+02*
	Std	1.72e-07	1.80e+01	5.38e-01	3.43e+01	5.61e-01	6.43e+01	6.95e-01	6.04e+01
S6	Min	**6.35e-09**	**0.00e+00**	1.94e+01	1.23e+01	*2.77e+00*	1.52e-01	2.92e+00	*1.16e-01*
	Mean	**2.00e-07**	**6.08e-10**	1.99e+01	1.70e+01	4.52e+00	9.81e-01	*4.25e+00*	*8.62e-01*
	Std	2.34e-07	3.18e-09	1.11e-01	2.78e+00	8.65e-01	4.82e-01	9.38e-01	4.84e-01
S7	Min	**4.40e+01**	**0.00e+00**	*4.89e+01*	3.65e+01	8.27e+01	2.49e+01	1.15e+02	*2.39e+01*
	Mean	**4.43e+01**	**0.00e+00**	*1.59e+02*	2.02e+02	2.76e+02	4.89e+01	2.76e+02	*4.45e+01*
	Std	1.18e-01	0.00e+00	4.42e+01	7.67e+01	1.18e+02	1.61e+01	1.19e+02	1.88e+01
S8	Min	**2.94e-12**	**5.69e-03**	*1.28e-02*	1.89e+01	1.13e-01	*4.68e+00*	3.70e-02	5.84e+00
	Mean	**3.07e-08**	**5.08e-01**	*3.26e-02*	2.43e+01	3.84e-01	9.66e+00	2.46e-01	*8.89e+00*
	Std	5.67e-08	7.83e-01	1.23e-02	2.92e+00	1.99e-01	1.74e+00	1.59e-01	1.49e+00
S9	Min	*5.78e+01*	**6.36e-04**	3.26e+02	1.78e+03	3.04e+02	6.72e-04	**2.00e+00**	*6.45e-04*
	Mean	**9.13e+01**	**6.36e-04**	5.02e+02	2.72e+03	5.02e+02	*1.41e+03*	*4.71e+02*	1.44e+03
	Std	2.97e+01	0.00e+00	8.46e+01	5.42e+02	1.18e+02	8.24e+02	1.51e+02	1.00e+03
w/s/l	Min	8/1/0	9/0/0	0/2/7	0/1/8	0/3/6	0/2/7	1/3/5	0/6/3
	Mean	9/0/0	9/0/0	0/2/7	0/1/8	0/0/9	0/2/7	0/5/4	0/6/3

TABLE IV

PERFORMANCES OF MTO-DRA, MFEA AND MFDE ON SET 19 PROBLEMS. THE BEST RESULTS BETWEEN TWO ALGORITHMS ARE TYPED IN **BOLD**.

Set 10	Alg	T_1	T_2	T_3
Mean	MTO-DRA	**1.5277e-15**	**3.8346e-15**	**0.0000e+00**
	MFEA	3.1926e+00	3.3428e+01	9.2668e+01
	MFDE	2.9748e+00	2.0891e+01	4.3984e+01
Min	MTO-DRA	**8.8818e-16**	**2.3785e-18**	**0.0000e+00**
	MFEA	1.8060e+00	2.5757e+01	3.6235e+01
	MFDE	1.9979e+00	1.5654e+01	1.8367e+01

TABLE V

PERFORMANCES OF MTO-DRA, MFEA AND MFDE ON SET 20 PROBLEMS. THE BEST RESULTS BETWEEN TWO ALGORITHMS ARE TYPED IN **BOLD**.

Set 11	Alg	T_1	T_2	T_3
Mean	MTO-DRA	**8.0557e+01**	**1.1886e+01**	**2.9638e+01**
	MFEA	2.2251e+02	3.3790e+01	1.3412e+02
	MFDE	2.4135e+02	2.2216e+01	8.7378e+04
Min	MTO-DRA	**2.3879e+01**	**4.5789e-08**	**6.2763e+00**
	MFEA	1.1233e+02	2.7449e+01	5.4702e+01
	MFDE	1.3528e+02	1.8459e+01	9.9587e+03

[4] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, 2016.

[5] M. Iqbal, W. N. Browne, and M. Zhang, "Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 465–480, 2014.

[6] R. Mills, T. Jansen, and R. Watson, "Transforming evolutionary search into higher-level evolutionary search by capturing problem structure," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 628–642, 2014.

[7] L. Feng, Y. S. Ong, M. Lim, and I. W.-H. Tsang, "Memetic Search with Inter-Domain Learning: A Realization between CVRP and CARP," *IEEE Trans. Evol. Comput. (CEC)*, vol. 19, no. 5, pp. 644–658, 2015.

[8] L. Feng, Y. S. Ong, M. H. Lim, and I. W. Tsang, "Memetic Search with Interdomain Learning: A Realization between CVRP and CARP," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 644–658, 2015.

[9] Y. Yuan, Y. S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with tsp, qap, lop, and jsp," in *IEEE Region 10 Conf. (TENCON)*, Singapore, 2016.

[10] A. Gupta, Y.-S. Ong, L. Feng, and K.-C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, 2017.

[11] C. R. Cloninger, J. Rice, and T. Reich, "Multifactorial inheritance with cultural transmission and assortative mating. II. a general model of combined polygenic and cultural inheritance." *Am. J. Hum. Genet.*, vol. 31, no. 2, p. 176, 1979.

[12] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, and T. P. Siew, "Linearized domain adaptation in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1295–1302.

[13] R. Sagarna and Y. S. Ong, "Concurrently searching branches in software tests generation through multitask evolution," in *IEEE Symp. Series Comput. Intell. (SSCI)*, 2016.

[14] G. Li, Q. Zhang, and W. Gao, "Multipopulation evolution framework for multifactorial optimization," in *Proc. Genet. Evolut. Comput. Conf. Comp.* Kyoto, Japan: ACM, 2018.

[15] R. Storn and K. Price, "Differential Evolution A Simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[16] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 2, 2005, pp. 1785–1791.

[17] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, 2009.

[18] J. Zhang and A. C. Sanderson, "JADE : Adaptive Differential Evolution with Optional External Archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[19] 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," Nanyang Technol. Univ., Tech. Rep., 2016.

[20] L. Feng, W. Zhou, L. Zhou, S. W. Jiang, J. H. Zhong, B. S. Da, Z. X. Zhu, and Y. Wang, "An empirical study of multifactorial PSO and multifactorial DE," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 921–928.