Multi-Task Particle Swarm Optimization With Dynamic Neighbor and Level-Based Inter-Task Learning

Zedong Tang, Maoguo Gong, *Senior Member, IEEE*, Yu Xie, Hao Li, and A. K. Qin, *Senior Member, IEEE*

***Abstract*—Existing multifactorial particle swarm optimization algorithms treat all particles equally with a consistent inter-task exemplar selection and generation strategy. This may lead to poor performance when the algorithm searches partial optimal areas belonging to different tasks at the later stage. In pedagogy, teachers teach students in different levels distinctively under their cognitive and learning abilities. Inspired by this idea, in this work, we devise a novel level-based inter-task learning strategy upon a dynamic local topology of inter-task particles. The proposed method separates particles into several levels and assigns particles to different levels with distinct inter-task learning methods. Specifically, we propose a level-based inter-task learning strategy to transfer sharing information among the cross-task neighborhood. By assigning the particles with diverse search preferences, the algorithm is able to explore the search space by using the cross-task knowledge, while reserving an ability to refine the search area. In addition, to address the issue of inter-task neighbor selection, we reform dynamically the local topology structure across the inter-task particles by methodical sampling, evaluating and selecting processes. Experimental results on the benchmark problems demonstrate that the proposed method enables the efficient cross-domain information transfer via the level-based inter-task learning.**

***Index Terms*—Evolutionary multitasking, multifactorial optimization, multitask optimization, particle swarm optimization.**

I. Introduction

EVOLUTIONARY multitasking is an emerging topic in the field of optimization, and attracts a wide attention of the researchers. Evolutionary multitasking [1] aims at leveraging the implicit parallelism featured by evolutionary algorithms (EAs) [2], [3] to solve multitask optimization problems (MTOs). From a theoretical point of view [4], it is possible to leverage the underlying complementarity between multiple tasks in implicit or explicit measures, thus accelerate the search for global optimums of multiple tasks. Actually, the evolutionary multitasking engine, which can minimize makespan of several optimization tasks, can be envisioned to be a heart of a cloud-based on-demand optimization system which faces multiple jobs submitted from enormous users.

Currently, multitask optimization problems are usually studied under the paradigm of multifactorial optimization (MFO) [5] where every component task in MTO acts as a contributed factor influencing the evolution of individuals. For simulating the evolution of individuals in multifactorial environment, multifactorial evolutionary algorithm (MFEA) with a computational counterpart of the multifactorial inheritance [6] was proposed in [5]. In MFEA, the building blocks corresponding to various tasks are encompassed in the pool of unified genetic materials. MFEA exploits the implicit parallelism of the population-based search methods to navigate multiple search spaces simultaneously. Although, each individual in the population focuses on a specific task in MFEA, individuals assigned with different tasks can exchange sharing genetic materials. Due to the appropriate exploitation of shared problem-solving experience, MFOs can attain a superior convergence property and enhanced accuracy in solving related optimization problems. Nowadays, these methods have attracted increasing attention for its potential to address some challenging problems [7]–[11].

As particle swarm optimization (PSO) has achieved a great success in solving continuous problems [12]–[17], it is natural to adapt them to the MFO paradigm like in [18][19]. Taking a close observation on the existing multifactorial particle swarm optimization algorithms (MFPSOs) [18]–[20], it is the fact that they usually treat all particles equally through the entire search process. Inflexible and nonadaptive selection and generation strategies of inter-task exemplars employed therein can result in the poor performance of refined search. We summarize the drawbacks of the previous MFPSOs below,

- Fully connected structure is not efficient at preserving the local features of multiple landscapes which might benefit solving these tasks.

Manuscript received March 2, 2020; revised June 15, 2020, September 2, 2020, and December 4, 2020; accepted December 31, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 62036006, in part by the National key research and development program of China uner Grant 2017YFB0802200, and in part by the Key Research and Development Program of Shaanxi Province under Grant 2018ZDXM-GY-045. *(Corresponding author: Maoguo Gong.)*

Zedong Tang is with the Academy of Advanced Interdisciplinary Research, Xidian University, Shaanxi 710071, China (e-mail: omegatangzd@gmail.com).

Maoguo Gong and Hao Li are with the School of Electronic and Engineering, Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Shaanxi 710071, China (e-mail: gong@ieee.org; omegalihao@gmail.com).

Yu Xie is with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China (e-mail: sxlljcxy@gmail.com).

A. K. Qin is with the Department of Computer Science and Software Engineering, Swinburne University of Technology, John Street, Hawthorn, Victoria 3122, Australia (e-mail: kqin@swin.edu.au).

This article has supplementary downloadable material available at https://doi.org/10.1109/TETCI.2021.3051970, provided by the authors.

Digital Object Identifier 10.1109/TETCI.2021.3051970

- There is a coefficient weighting the inter-task acceleration term, which is sensitive to problems. For users tuning this parameter is challenging since other acceleration terms can influence it.
- Each particle has different search potentials in the inter-task learning process. It is promising to enhance the stability of inter-task learning by developing particles' potentials.
- Random selecting the inter-task leader will cause unexpected perturbation into the swarm. Besides, there is no protection of the good particles from being perturbed by inter-task learning, which leads to the algorithms inclining to exploration.

To overcome the drawbacks of MFPSO, we propose an improved multifactorial PSO (DNL-MFPSO) algorithm by introducing a local dynamic topology among the cross-domain particles, along with a new level-based inter-task knowledge transfer scheme. The local dynamic topology structure is motivated by the recent researches of the local versions of PSO for multimodal problems [21]–[24]. The local PSO can especially preserve high diversity to avoid premature convergence, which potentially benefits the exploitation of inter-task search experience. For robustness, we propose a periodic inter-task neighbor reforming to select favorable inter-task neighbor for particles, instead of completely random sampling. By this means, a favorable inter-task neighbor leads the particle in a relatively long period and avoid being misleading by multiple random neighbors changing in a short period. To aid the synthesis of intra-task and inter-task neighbors, we present a new level-based inter-task learning strategy. Specifically, the particles are partitioned into different levels according to their search potentials and can choose distinctive learning strategies based on their levels. This inter-task learning strategy is inspired by the idea that in pedagogy teachers usually teach students according to their cognitive and learning abilities. The students should be distinctively treated in accordance of their aptitudes [25]. In the proposed algorithm, the particles can choose their learning methods based on the their search potentials. Besides, ensemble learning methods with unique properties can be an effective way to tackle with the knowledge transfer among various types of problems.

The major contributions are summarized as follows,

- To full use the knowledge induced in the local features of different decision spaces, a dynamic inter-task neighbor reforming strategy is proposed.
- Inspired by level-based pedagogy, we propose a level-based inter-task learning strategy to promote the particles' searching potential.
- We design three learning methods associated with different levels.
- The proposed method is testified in the synthetic and practical problems, and shows its superior performance.

The rest of this paper is organized as follows. Section II describes the background related to this research. The proposed MFPSO algorithm is elaborated in Section III. In Section IV, the efficacy of the proposed method is testified by conducting empirical experiments. More experimental results are listed in supplementary materials. In Section V, we conclude the paper with some future works mentioned.

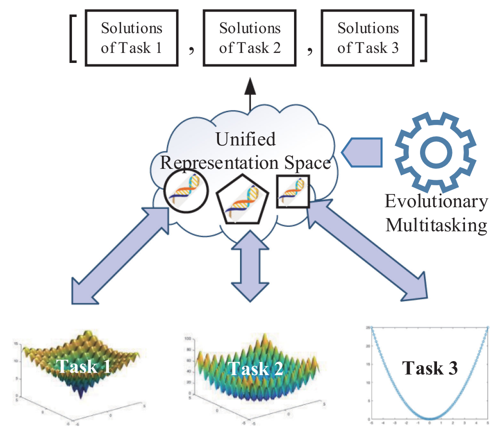


Fig. 1. In MFO, variables of various tasks can be encoded into a unified representation scheme. The MFO algorithms act as a multiple-input and multiple-output model, which can optimize multiple tasks simultaneously and output their solutions.

II. Prelimitaries and Related Work

A. Multifactorial Optimization

MTO aims to promote the transfer of omnidirectional knowledge to assist the efficient resolution of multiple tasks with various properties. Without loss of generality, a minimization problem of MTO with K tasks is formulated explicitly as $\{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_K^*\} = \arg\min\{f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \quad \ldots, f_K(\mathbf{x}_K)\}$. Evolutionary multitasking framework was put forward in [7], where the nature-inspired population-based methods are employed to solve MTO problems by unleashing the potential of the implicit parallelism of population-based search algorithms. As suggested in [26], a multitask optimization problem and the corresponding multitask evolutionary algorithm are denoted as multifactorial optimization (MFO) and multifactorial evolutionary algorithm (MFEA), respectively.

In MFO, each f_i among the composite problem can be particularly viewed as an existing factor influencing the evolution of individuals in the K-factorial environment [6]. First, as illustrated in Fig. 1, the specific tasks' search spaces are mapped into a unified representation space. The uniform random key scheme [5] is a popular method which maps the solutions assigned to different decision spaces into a unified decision space. Then, the MFO algorithms can use a single population to navigate the unified representation space by a consistent means. MFO features assortation mating and selective imitation. Assortation mating is a key characteristic which enables the implicit transfer of shared problem-solving knowledge. The later feature allows the offspring to imitate the phenotype of its parent and enables the transfer of cultural traits. Meanwhile, the selective imitation allows each individual being only evaluated on task associated to it for the sake of reducing the computational efforts.

To compare the individuals' performances, there are some parameters associated with the individuals.

1) *Factorial Cost [26]:* The factorial cost of individual x_i on task T_j is the corresponding objective value of potential solution x_i given by ψ_i^j.

2) *Factorial Rank [26]:* The factorial rank of x_i on T_j is the rank index of x_i in the sorted objective value list given by ϕ_i^j.
3) *Scalar Factor [26]:* The skill factor is defined by the index of the task which an individual is assigned to. Skill factor of x_i is given by $\tau_i = \text{argmin}_{j\in\{1,2,\ldots,K\}}\phi_i^j$.
4) *Skill Fitness:* The scalar fitness [26] of x_i is the inverse of the minimal ϕ_j^i given by $\theta_i = 1/\min_{j\in\{1,\ldots,K\}}\phi_i^j$.

The skill factor is regarded as the cultural trait of the individual explicitly. The scalar fitness is viewed as a criterion of individual's performance in multifactorial environment. A popular implementation of MFO is MFEA whose basic structure of MFEA can be found in [26].

B. Particle Swarm Optimization

PSO [27] is a population-based stochastic optimization algorithm which is relatively new compared with other evolutionary computation methods and has been used for many real-world applications successfully. In fact, PSO is meta-heuristically inspired by the social behavior of bird flocking and fish schooling. Henceforth, the population in PSO is called "swarm". Each potential solution, called particle, is associated with a position and a velocity. PSO adjusts iteratively the velocity of a particle by integrating its best historical position and the best position found by the neighbors of the particle. The velocity v_{id} and position x_{id} of the dth dimension of the ith particle are updated as follows [28]:

$$v_{id} \leftarrow K * [v_{id} + c_1 * \text{rand}() * (p_{id} - x_{id}) + c_2 * \text{rand}() * (p_{gd} - x_{id})], \tag{1}$$

$$K = \frac{2}{|2 - c - \sqrt{\gamma * (c-4)}|}, \text{where } c = c_1 + c_2 \geq 4.1, \tag{2}$$

$$x_{id} \leftarrow x_{id} + v_{id}, \tag{3}$$

where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ represents the position of the ith particle, and $\mathbf{v}_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ represents the velocity assigned to the ith particle. The equation (1) comprises three terms, momentum, cognitive and social acceleration terms [28]. c_1 and c_2 are two positive acceleration coefficients, which influence the weightings of acceleration terms. In the cognitive acceleration term, $\boldsymbol{p}_i = [p_{i1}, p_{i2}, .., p_{iD}]$, $\boldsymbol{p}_i \in P_m$ is the history best position of particle i. In the social terms, $\boldsymbol{p}_g = [p_{g1}, p_{g2}, \ldots, p_{gD}]$, $\boldsymbol{p}_g \in P_g$ is the best position found by the neighbors of particle i in the last generation. Each particle's velocity on each dimension is clamped to a maximum velocity magnitude $V_{\max}$. If $|v_i^d|$ exceeds $V_{\max}$, which is a parameter specified by the user, the velocity of that dimension is limited to $V_{\max}$.

As per the social structure of the neighborhood of a particle, PSO can be roughly clustered into two variants, global PSO and local PSO, known as $gbest$ PSO and $lbest$ PSO. In the canonical global PSO algorithm, an individual updates its position and velocity according to the personal best position and the global best position of the whole swarm. This can be regarded as a particle's learning action from its owned experiences and social experiences. The learning-based strategy enables the fast propagation of information between the particles, thereby the swarm converges fast. Moreover, the convergence speed can be controlled by devising a more specified learning strategy and modifying the social structure of particles. Several variants of local PSO designing some efficient learning strategies have been developed currently [29]–[31]. DMSPSO [32] is an improved PSO with a random neighbor structure which can highly reserve the diversity of swarm. Level-based PSO [33] can be viewed as a local PSO, where the particles are partitioned into several levels, and the exemplar is only selected from higher levels than that of each particle.

C. Related Work

Since the efficiency of MFO, increasing research interests in developing MFO algorithms to solving various problems have risen. Yuan *et al.* [8] proposed an exquisite 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. Sagarna *et al.* [9] applied MFEA on the software tests generation. This work mainly concentrates on branch searching and is the first time to apply MFEA to real-world problems with more than two tasks. Gupta *et al.* [7] proposed a meaningful variant MFEA for multitasking multi-objective. A realization of an off-the-shelf evolutionary multitasking paradigm forward multi-objective optimization was represented in that work.

Since the primary framework was proposed in [26], many enhancements on MFO have been investigated. Some works focus on improving the efficiency of MFEA. Bali *et al.* [34] designed a linear domain adaptation and obtained a high order representative space to alleviate the negative knowledge transfer between uncorrelated tasks. Wen *et al.* [35] analyzed the behavior of cross-domain crossover and proposed a novel strategy to balance the inter-domain and inner-domain crossover. Feng *et al.* [36] proposed an evolutionary memetic computing paradigm which is capable of learning and leveraging the inter-domain knowledge of two kinds of assignment problems. Bali *et al.* [4] proposed an evolutionary multitask framework, called MFEA-II, where the similarities between tasks are computed online based on Kullback-Leibler divergence. Then the random mating probability matrix is learned for the adaptive information transfer in MFEA-II. Besides, there are some works devoting to incorporate the other population-based search methods into the MFO paradigm. Gupta *et al.* [7] adapted MFEA in the field of multiobjective optimization by employing the nondomination rank as the performance criteria and proposed a multifactorial multiobjective optimization paradigm (MFMOO). Feng *et al.* [18] incorporated the swarm intelligence method including particle swarm optimization (PSO) and differential evolution (DE) into MTO towards multifactorial particle swarm optimization (MFPSO) and multifactorial differential evolution (MFDE). Wen and Ting [37] incorporated the MFEA paradigm into genetic programming, leading to the MFGP algorithm for learning the ensemble of classifiers.

Some works focus on developing the many-task optimization algorithms. Liaw *et al.* [38] made an early attempt to solving many-task optimization by simulating evolution of biocoenosis through symbiosis. This method can solve many-task optimization consisted of up to 50 tasks. To address the drawback of their previous work that lacks the ability of seeking the tasks to transfer the complementing information, Liaw *et al.* [39] designed a novel adaptive information transfer for many-task optimization. Chen *et al.* [40] proposed a novel evolutionary framework for many-task optimization, where a composite complementary measure based on the similarity between tasks and the accumulated rewards of knowledge transfer is utilized.

Besides, Liu *et al.* [41] proposed a surrogate-assisted multi-task optimization framework, where the population is split into multiple subpopulations and every subpopulation focuses on solving a task. Each subpopulation employs a surrogate-assisted approach with gaussian process to search the decision space. Chen *et al.* [42] incorporated an adaptive local search strategy into the multi-population multitask optimization framework for fast solving multi-task multi-objective problems. Huang *et al.* [43] proposed an efficient surrogate-assisted multi-task evolutionary framework called SaEF-AKT for tackling with the computationally expensive tasks. In addition, an adaptive knowledge transfer based on the tasks' similarity measure is employed in SaEF-AKT.

Since the learning strategy of the canonical PSO algorithm enables the transfer of knowledge through the swarm, the transfer of shared knowledge across multiple optimization tasks could be also realized by designing an inter-task learning strategy. An efficient design of the inter-task learning strategy can facilitate the implicit transfer of candidate solutions, leading to the variants of PSO for multifactorial optimization. MFPSO was first proposed in [18] where multiple swarms accounting for different optimization tasks can share their global best leaders. Afterwards, an improved MFPSO with inter-task learning was proposed in [20] where the whole swarm is dynamically separated into several subswarms each of that focuses on solving a task. Then, inter-task learning strategy generates a leader for each particle by synthesizing the inter-task and intra-task information. MFPSO can be considered as a dynamic system. The large momentum caused by the distant exemplars from another task with optimal solution will dramatically influence the stability of the algorithm. Therefore, the accuracy of final solutions might be significantly influenced by such exemplars.

III. Proposed Method

To ameliorate the issues of existing MFPSOs, we propose a new dynamic local topology PSO with level-based inter-task learning strategy (DNL-MFPSO). In this section, a new dynamic neighborhood structure exclusive to MFO is first established to aid the proposed MFPSO. With such a mechanism, DNL-MFPSO can maintain satisfactory solutions of multiple tasks, at the same time, without hampering the ability of exploring the complementary among tasks. And then, accompanying with this dynamic neighborhood structure, a level-based inter-task learning strategy is presented to aid the knowledge transfer.

TABLE I
The Search Preferences of Individuals and the Inter-Task Transfer Strategies

Level	Search Preference	Inter-task Transfer Strategy
1	Refined (Protected Elites)	Greedy
2	Moderately Explore	Linear Combination
3	Explore	Different Mutation

A. Basic Ideas

The local structure of swarm is usually used to control the selection of leaders and manage the convergence speed, aiming to comprise the balance of the exploration and exploitation. When adapting PSOs to MFO, selection of inter-task leaders and reservation of local features of multiple optimization problems' landscapes become essential for inter-task knowledge transfer. To address these issues, we propose a way to form the cellulars (a set of overlapping subsets of particles covering the whole swarm) among swarms such that the local features of tasks' landscapes can be reserved within the cellulars. Then, the inter-task neighbor of each particle is selected from the inner-task neighbors of a particle assigned to another task. A sampling-evaluation-selection technique is employed to save the enormous computation efforts. We design a dynamic reformation for multifactorial PSO along with two induced structures, intra-task cellulars and inter-task neighbors,

- A periodic dynamic multiple swarm strategy is employed to form cellulars in swarm.
- An inter-task neighbor of a particle is selected from a foreign cellular by a reevaluation-and-selection technique.

Another essential issue is the design of inter-task learning strategy for MFPSO. This is, how the particle having inter-task neighbor to merge two aspects of search experience, namely the historical personal best position and inter-task neighbor's position. Note that, during the evolution particles in different states usually have different potentials in inter-task exploration and intra-task refined search. Thus, they should be treated distinctively. In the proposed DNL-MFPSO, the particles are partitioned into different levels based on their performances and associated with specific inter-task learning methods to develop their potentials. Each transfer operation implies different interaction degree between inter-task particles in the inter-task learning procedure. To efficiently use the knowledge from different tasks, each particle can select a transfer operation according to its level. Table I demonstrates the search preferences of individuals and the corresponding inter-task learning behavior which individuals can choose. The purpose of the level-based inter-task learning is to utilize the inter-task information without harming the refined search. The technical details are exhibited in Section III-C.

B. Dynamic Local Topology Exclusive to MFO

1) Information Isolation in Evolutionary Multitasking: For a composite population tackling several tasks simultaneously, the compromise between isolation and mixture of building blocks is essential. The mixture of building blocks can help the individuals leverage the sharing knowledge to highly improve the efficiency

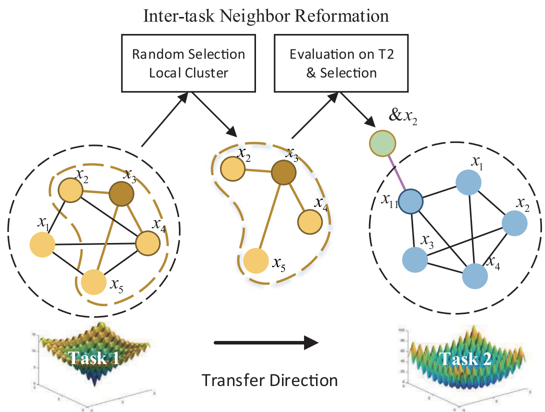


Fig. 2. Dynamic topology of particles which the inter-task connections are sparse and the intra-task connections are dense. The expected number of particles assigned with inter-task neighbors is equal to $rlp \cdot N_p$.

of problem solving and can preserve the diversity of the population. However, for tasks with distant optimal solutions, the negative transfer is a pressing issue. Multitask optimization methods suffer a poor performance of refining the search areas on these tasks. So, there is another topic on information isolation in the field of multitask optimization.

A proper isolation strategy is rather important, which can manage the propagation of cross-domain information and reduce the negative transfer to avoid unnecessary perturbation of the population. MFPSO was proposed as a MFO version of PSO where the cross-domain knowledge is exchanged via the velocity updating method. Owing to the absence of selection operation, MFPSO relies completely on the leader selection and velocity updating to drive the swarm. However, since MFPSO only uses the global best positions of foreign tasks in the inter-task learning as leaders, the large perturbation can influence the particles for a period, dependent on the inertia weight implying the half-life of the current momentum. This increases the opportunity for the occurrence of negative transfer. Despite there are some efforts having been made to find a proper isolation strategy for MFEA by employing multiple population techniques [44]–[46], seldom has an effort been made toward exploring the topology structure for MFPSO to efficiently manage the information exchange. Moreover, except for global best particles, MFPSO could not make full use of the knowledge of local features embedding in the landscapes of different tasks.

2) The Proposed Local Structure for MFPSO: In this study, we focus on dynamically reforming the inter-task local neighborhood structure to control the inter-task information propagation. The procedure of reforming the particles' inter-task neighbors can be depicted in Fig. 2. Assume that there are N_p particles for solving K tasks. The proposed method can establish K subswarms $\{A_k, k = 1, \ldots, K\}$ for K tasks by random initialization. Each subswarm assigned to one task has N_p/K particles. Every particle $\boldsymbol{x}_i, i = 1, \ldots, N_p/K$ in A_k has a skill-factor $\tau(\boldsymbol{x}_i)$. The subswarm structure guarantees the isolation of task-specified information. At the same time, the cross-domain transfer of the problem-solving knowledge is exchanged through

---

Algorithm 1: Dynamic Local Topology Strategy.

Input: Tasks $\{T_1, \ldots, T_K\}$, Subswarms $\{A_1, \ldots, A_K\}$
Output: Intra-task neighbors $\{B_i\}$, inter-task neighbors $\{\boldsymbol{x}_{ti}\}, i = 1, \ldots, N_p$

```
For Each task τ do
    If no improvement is achieved within G generations then
        For x_i in A_τ do
            If rand() < rlp then
                Select a learned particle x'_t for ∪_{k≠τ}{A_k};
                Evaluate the x'_t and its neighbors B_t on τ;
                Select a inter-task neighbor x_t for x_i by using
                elite selection method in B_t ∪ {x'_t};
                Choose the N_t intra-task neighbors in A_τ for
                x_i and store in B_i;
            Else
                Choose arbitrary N_t particle of the same task
                as the intra-task neighbors of x_i and store in
                B_i;
            End If
        End For
    End If
End For
```

---

the inter-task neighborhood. By this means, the inter-task information exchange between can be limited among the particles being connected in the inter-task neighbor reformation process. Good position will propagate through the corresponding intra-task connections that are denser than inter-task ones, once it is found with the help of the inter-task neighbor. This suggests fast propagation of good solutions within the subswarm, but a managed inter-task information exchange across subswarms assigned to optimization tasks. The proposed method ensures that only the strongly favorable information can pass through the narrow paths connecting two different tasks. The number of particles having inter-task neighbors is proportional to the random learning parameter rlp. The inter-task neighbor reformation process is adaptively executed. If the subswarm of a task is stagnated in a fixed period, the connections between subswarms are reformed dynamically by a sampling-evaluating-selecting strategy. For example, we determine the inter-task neighbor of $\boldsymbol{x}_i$ belonging to $T_{\tau(\boldsymbol{x}_i)}$. We first choose a task T_r except for the current task $T_{\tau(\boldsymbol{x}_i)}$. Then, a particle $\boldsymbol{x}'_t$ and its neighbors B_t are selected as candidates for the leader designations and evaluated on $T_{\tau(\boldsymbol{x}_i)}$. The fittest one is selected via elite selection in terms of objective function values corresponding to $T_{\tau(\boldsymbol{x}_i)}$ as the inter-task neighbor of $\boldsymbol{x}_i$. The proposed method is shown in Algorithm 1.

3) The Construction of Local Social Structure: The proposed DNL-MFPSO constructs and maintains the neighbor topology structure during the evolutionary process. When the swarm get stagnated for G generations, the proposed dynamic neighbor reformation is activated for establishing the intra-task and inter-task neighbors of every particle. Each particle has a set B_i of N_t intra-task neighbors and an inter-task neighbor $\boldsymbol{x}_t$. As we focus on the particle denoted as $\boldsymbol{x}_i$ and the subswarm that $\boldsymbol{x}_i$ belongs to denoted as $A_{\tau(\boldsymbol{x}_i)}$, the neighborhood adjusting strategy is elucidated as follows.

- *Selection of intra-task neighbors:* Therein, N_t neighbors of $\boldsymbol{x}_i$ are sampled randomly from $A_{\tau(\boldsymbol{x}_i)}$ without replacement, denoted as B_i. By this means, the proposed method generates that every individual has a chance to become the neighbors each other in the whole process of evolution, thereby, the search diversity of the subswarm can be reserved.
- *Selection of inter-task neighbors:* The inter-task neighbor is determined by employing a sampling-then-selecting method. First, randomly selecting a task T_r in $\{1, \ldots, K\}$ s.t. $T_r \neq \tau(\boldsymbol{x}_i)$. A particle $\boldsymbol{x}'_t$ is randomly selected from A_{T_r}. Then, the neighbors B_t of $\boldsymbol{x}'_t$ in T_r are evaluated on the task $\tau(\boldsymbol{x}_i)$. An elite selection method is utilized to select one particle within the candidate particles as the inter-task neighbor $\boldsymbol{x}_t$ of $\boldsymbol{x}_i$. The process can be formulated as follows,

$$\boldsymbol{x}_t = \text{arg min}_{\boldsymbol{y} \in B_t \cup \{\boldsymbol{x}'_t\}} f_{\tau(\boldsymbol{x}_i)}(\boldsymbol{y}) \tag{4}$$

  where $f_{\tau(\boldsymbol{x}_i)}$ is the fitness function corresponding to $\tau(\boldsymbol{x}_i)$. This aims at selecting the neighbor, which is likely to generate a promise solution combined with $\boldsymbol{x}_i$, from the other tasks for each particle. This reveals the influence of negative transfer on the search performance.

As for the initialization of inter-task and intra-task neighbors, we simply select the neighbors randomly in whole swarm. Last but not least, it is worthwhile to notice that the indices are stored into the neighborhood pool, instead of the current positions of them, such that the particles can use the neighbors' current positions mutable in every moment when the learning strategy is executed.

4) Discussion: The proposed dynamic neighbor construction procedure spends most of computational cost in the selection of inter-task neighbors. Briefly, the computational complexity of a single execution can be calculated by counting the number of objective function evolutions. According to **Algorithm 1**, there are $N_p \times rlp$ individuals having inter-task neighbors in the current generation. To select an inter-task neighbor for every particle, we need to evaluate the corresponding inter-task particle and its neighbors from foreign tasks, which requires N_t calls of the objective function. Thus, the computational complexity of a single run of dynamic neighbor construction procedure is $O(N_p \cdot N_t \cdot rlp)$. The settings of parameters N_t and G are analyzed by empirical experiments. The detail results are shown in the experimental section.

C. Level-Based Hybrid Inter-Task Learning

Taking a close look at the framework of PSOs, these algorithms rely on the exemplars to mutually exchange information. The memory updating, including social and personal best position updating, replaces the role of natural selection in GAs. Thereby, we consider utilizing the dynamic social structure among the swarm to isolate the cross-domain information and, at the same time, enable the inter-task knowledge exchange by the inter-task neighbors. Hereafter, we mainly focus on designing the level-based inter-task learning strategy which synthesizes the inter-task and intra-task neighbors to generate the exemplars.

Tim O'Brien *et. al.* [25] suggested that the students should be distinctly taught to develop their potentials. Similarly, in MFO, the particles associated with the same task can particularly have different search preferences. The particles close to the optimal area are mainly focused on refining the search area. In contrast, particles scatted in a broad area devote to explore the search space. In order to preserve the good solutions, at the same time, leveraging on the complementary among tasks simultaneously, it is promising to associate the particles with distinctive learning methods in accordance with their search preferences. Thus, we proposed a level-based inter-task learning method to generate inter-task exemplar. Specifically, the N_p particles in each subswarm are equally partitioned into N_L levels denoted by $L_i (i = 1, \ldots, N_L)$ by ranking their fitnesses, and we assign them with distinct inter-task learning methods, denoted by **Method *L***, with the unique search characteristics. Such that, level L_1 can particularly protect the elite particles settling in there by employing a conservative learning method. The particles in bottom levels are encouraged to search broad areas by employing learning methods with distinct degrees of exploration. The details of this strategy are provided in above content.

1) Inter-Task Exemplar Generating Strategies: We actualize the learning strategy with $N_L = 3$. For clarity, three learning methods for Level 1, **Level 2** and **Level 3** are respectively denoted as **Method 1**, **Method 2** and **Method 3**. We discuss the specific formulas and the features of the corresponding learning methods as following. First, the denotations are defined. $\boldsymbol{x}_i$ is the current particle. $\boldsymbol{p}_i$ is the history best position of $\boldsymbol{x}_i$. $\boldsymbol{x}_t$ is the inter-task neighbor of $\boldsymbol{x}_i$. $\boldsymbol{p}_t$ is the personal best position of the inter-task neighbor $\boldsymbol{x}_t$. $\tau(\cdot)$ represents the skillfactor of a particle. $\boldsymbol{p}_n \in B_i$ denotes the local best of the particle $\boldsymbol{x}_i$. $\boldsymbol{p}_t$ denotes the inter-task neighbor's personal best position. Afterwards, we formulate the three learning methods, as listed below.

- *Level 1. A greedy strategy is to select the better exemplar:* The fitnesses of the current position x_i and the inter-task neighbor personal best position $\boldsymbol{p}_t$ are compared. If $\boldsymbol{p}_t$ is better than $\boldsymbol{x}_i$, then p_t is employed as its exemplar. Otherwise, the exemplar is $\boldsymbol{u}_i$. This method can be viewed as a very restricted transfer strategy, because it employs an exploiting-only strategy. Particularly, the exemplar is duplicated from the inter-task neighbor's best position or the local best position, it does not generate an exemplar with new position at all. By this way, the elite particles can be reserved as much as possible.

$$u_{id} = \begin{cases} p_{id} \text{ if } f_\tau(\boldsymbol{x}_i) > f_\tau(\boldsymbol{p}_t), \\ p_{td} \text{ otherwise.} \end{cases} \tag{5}$$

  Above method is deterministic, therefore focuses on the exploitation.
- *Level 2. The exemplar is generated by an arithmetic crossover:* This strategy encourages each particle exploring

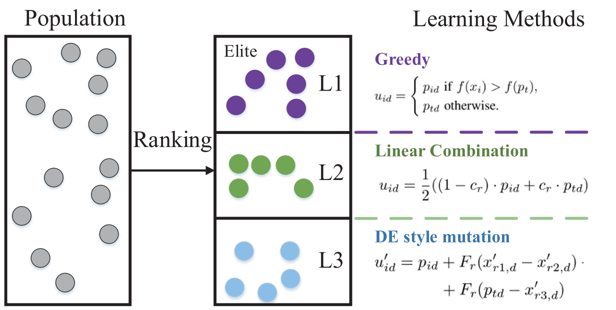


Fig. 3. The framework of the inter-task Level-based Learning strategy. First, particles in every subswarm are sorted in descending order of fitness and then they are equally divided into 3 levels (L1, L2 and L3). Then, Particles in L3 learn from the exemplars by mutating the local best position via residuals from the subswarm owning the inter-task neighbor. Particles in L2 learn from the exemplar generated by linearly combining the local best position and the inter-task neighbor. To protect, most promising particles, particles in L1 learn greedily from the better ones between local best position and inter-task neighbor position.

the areas between the each particle's history best $\boldsymbol{p}_i$ and the inter-task neighbor personal best p_t. This procedure generates the exemplar via synthesizing these positions shown as follows,

$$u_{id} = (1 - c_r) \cdot p_{id} + c_r \cdot p_{td} \tag{6}$$

where c_r denotes a randomly generated value in the range [0,1]. This method use randomness on the selection of c_r and $\boldsymbol{p}_t$ to enhance the diversity of inter-task learning.

- *Level 3. The exemplar is generated via a differential mutation operation:* A trial vector is first generated via differential operation. Herein, we adopt the DE/to-best/2 as specific differential operation.

$$u_{id} = p_{id} + F_r(x'_{r1,d} - x'_{r2,d}) + F_r(p_{td} - x'_{r3,d}) \tag{7}$$

where $\boldsymbol{x}'_{r1}$, $\boldsymbol{x}'_{r2}$ and x'_{r3} are drawn from subswarm owning to. F_r is a uniformly distributed number in the range of $[0, 1]$. After that, the trial vector crossovers with the current particle to construct the candidate exemplar as shown in following formula.

$$u_{id} = \begin{cases} u_{id} & \text{if } rand() < \phi_i^{\tau_i}/N_p, \\ p_{id} & \text{otherwise} \end{cases} \tag{8}$$

where N_p is the swarm size, and $\phi_i^{\tau_i}$ is the rank of individual $\boldsymbol{p}_i$ in terms of the τ_i objective function. Because this learning method generates the candidate exemplars embedding most randomness among three methods, this strategy has a great ability of exploration. Thereby, these exemplars counterbalance the local exploitation and helps in utilizing the underlying complementary of function landscapes to escape the local optimums.

The proposed method generates the exemplars of particles according to the particles' search preferences (exploration or refining). As depicted in Fig. 4, by using various generation operations, three strategies can generate the exemplars drawn from various regions. For the sake of analyzing the generated exemplar's influence in the search ability, we begin by neglecting the term of the personal memory in the velocity updating rule.

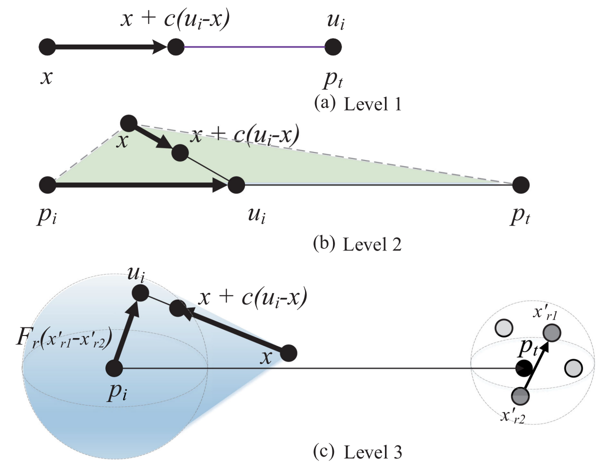


Fig. 4. The searching areas of three methods used in three level. (a) A greedy selection strategy ensures the restricted search area; (b) The synthesizing of inter-task local memory provides a broad search area; (c) The inter-task DE mutation is capable of synthesizing distributional knowledge. The shadow areas are the expected movements of particles in next step.

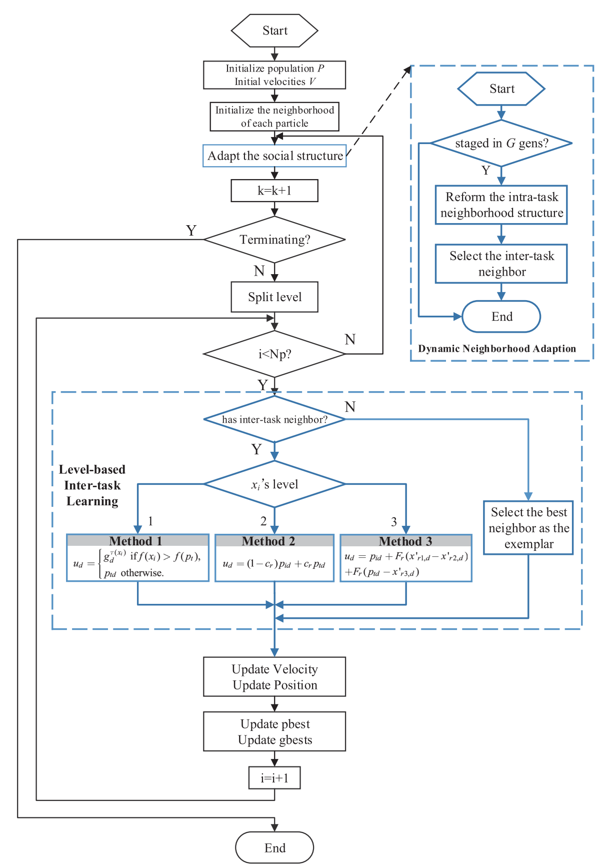


Fig. 5. Main framework of DNL-MFPSO.

Assume that we are to calculate the exemplar point of the particle $\boldsymbol{x}$. $\boldsymbol{p}_i$ and $\boldsymbol{p}_t$ are the personal best position of $\boldsymbol{x}$ and the inter-task neighbor's best position of $\boldsymbol{x}_t$ respectively.

Herein, we provide an intuitive analysis of three searching behavior. When the inter-task information occurs, **Method 1** treats the $\boldsymbol{p}_t$ as the leader of $\boldsymbol{x}$. The new position of $\boldsymbol{x}$ is distributed on the forward direction of the vector from $\boldsymbol{x}$ to $\boldsymbol{p}_t$. Thus, the particle can search a restricted area. It is worthwhile to note that **Method 1** is a greedy strategy when selecting the $\boldsymbol{u}_i$ from $\boldsymbol{p}_i$ and $\boldsymbol{p}_t$ by comparing the fitness. Therefore, **Method 1** confines the moving trajectory of a particle within the local area, although the inter-task experience is utilized. **Method 2** generates an exemplar $\boldsymbol{u}_i$ from a relatively broad region formed by the three points, $\boldsymbol{x}$, $\boldsymbol{p}_i$ and $\boldsymbol{p}_t$. If the acceleration parameter c in the velocity updating equation is a real positive number, the search region is a sector in the hyper plane formed by $\boldsymbol{x}$, $\boldsymbol{p}_i$ and $\boldsymbol{p}_t$. Therefore, **Method 2** can generate an exemplar $\boldsymbol{u}_i$ located in a wider region than **Method 1** do. At the third level, **Method 3** employs the differential mutation operation to transfer the distributional knowledge. It is characterized that the swarm information is utilized to ensure the exemplar being drawn from a region resembling to the landscape of the source subswarm. Consequently, by synthesizing the distributional knowledge, the particles are capable of escaping from the valleys of local optimums. **Method 3** can generate the exemplars $\boldsymbol{u}_i$ drawn from a hyper-cone with a conical tip located at $\boldsymbol{x}$. According to the analysis, the particles can have different search behaviors. The particles near best region conduct greedy inter-task learning to protect the elites. At the same time, other particles attempt to search the intermediate region between different tasks. The proposed method can efficiently achieve the balance of inter-task searching experience transfer and the refining search.

2) Velocity Updating Rule: Combining the above components together, the swarm evolves iteratively according to the update rules formulated as follows,

$$
\begin{aligned}
v_{id} \leftarrow K \times [v_{id} + c_1 * R_1 * (p_{id} - x_{id}) \\
+ c_2 * R_2 * (u_{id} - x_{id})] \qquad (9)
\end{aligned}
$$

$$
x_{id} \leftarrow x_{id} + v_{id} \qquad (10)
$$

where x_{id} is the position of the ith particle in the dth dimension and p_{id} is its personal best position. u_{id} is the synthesized exemplar. Equation (10) gets rid of the additional inter-task acceleration term, such that one does not have to tune three problem-dependent parameter with respect to the properties of the tasks, like in MFPSO.

Since each particle be assigned to only one task in the whole evolution processing, a particle only evaluates on the corresponding task and the fitness function is calculated by normalizing the original function values into the range of [0,1]. We use the min-max mapping as the mapping function. Keeping the fact that the fitter individual has the larger fitness value in mind, we calculate the fitness of a particle in Equation (11)

$$
f(\boldsymbol{x}) = \frac{\max_{\boldsymbol{x} \in A_\tau} \psi_\tau(\boldsymbol{x}) - \psi_\tau(\boldsymbol{x})}{\max_{\boldsymbol{x} \in A_\tau} \psi_\tau(\boldsymbol{x}) - \min_{\boldsymbol{x} \in A_\tau} \psi_\tau(\boldsymbol{x})} \qquad (11)
$$

where $\psi_\tau(\boldsymbol{x})$ is the original objective function of the task $\boldsymbol{x}$ belonging. A_τ is the set composed of three categories of the particles, including the ones that belong to the same task as $\boldsymbol{x}$, the personal best particles and the global best particles. If the current particle's scalar fitness is greater than its personal best position's, the personal best position would be updated by the current position.

Algorithm 2: Level-based Inter-Task Learning.

Intput: Subswarms $A_1, \ldots, A_K$, Intra-task neighbors $\{B_i\}$, inter-task neighbors $\{\boldsymbol{x}_{ti}\}$, $i = 1, \ldots, N_p$
Output: Exemplars $\boldsymbol{u}_i, i = 1, \ldots, N_p$.

```
M = N_p/(K × N_L);
L_j = {}, j = 1, ..., N_L; n = 0, j = 1;
Partition equally the particles into 3 levels.
for k = 1 to K do
    Sort the particles in the subswarm A_k in ascending order of fitness;
    Get the sorted index of particles π = {π_1, π_2, ..., π_NP};
    for π_i in π do
        if j > N_L then
            Break;
        end if
        if n < M then
            L_j = L_j ∪ x_i;
            n++;
        else
            n = 0, j++;
        end if
    end for
    Insert the rest of particles into the set of the last level ℒ_{N_L} in ℒ.
end for
Generate the exemplars as per the levels.
for each x_i in P do
    if x_i has a inter-task neighbor then
        Select the best intra-task neighbor from B_i as the exemplar u_i.
    else
        If x_i ∈ L_1 then
            Update the local best via Method 1;
        Else If x_i ∈ L_2 then
            Update the local best via Method 2;
        Else If x_i ∈ L_3 then
            Update the local best via Method 3;
        End If
    End If
    If rand() < 0.1 then
        Mutate u_i.
    End If
    Evaluate the new personal position on task corresponding to the skill factor only;
End For
```

D. Framework of DNL-MFPSO

At the beginning of the algorithm, the swarm is initialized by assigning each particle uniformly distributed position and Gaussian distributed velocity with the zero mean . Then, the particles are assigned to the tasks equally, and the task id of

TABLE II
PROPERTIES OF DIFFERENT GROUP OF TEST PROBLEMS

	Uni/Multi-modal	Dimensionality	Optima
Group 1	Various	Same	Same
Group 2	Various	Different	Same
Group 3	Various	Same	Different

each particle is stored in the property τ. After the initialization, in the main loop of the evolution, the inter-task and intra-task neighbors are first assigned to the particles according to the random inter-task learning probability by using **Algorithm 3** in lines 12–15. By this means, the swarm can form a relatively stable social structure to reserve the optimums of multiple tasks, and reserve the necessary diversity therein, which the inter-task learning can benefit from. The particles are clustered into different levels in the rank of the objective functions. According to the particles' search preference (denoted as the conception of level), the novelty of the proposed learning strategy is to adapt various operations to combine the inter- and intra-neighbors. Next the velocities and positions of the particles are updated by the modified level-based learning strategy. When evaluating the newly generated positions, the normalized objective fitness replaces the rank-based factorial fitness so as to simplify the calculation.

E. Related to Previous Work

We note that in [47] the authors presented a similar idea using CGA for MFO compared to the proposed inter-task neighbor reforming strategy. But, we have to emphasize that the difference between our method and MFCGA is two-fold:

1) MFCGA only replace the GA in MFGA by the CGA. In MFCGA, the cellar reformation strategy processes all particles belonging to different tasks as a single swarm by the same strategy in CGA. In a word, the individuals assigned to different tasks are tackled as if they are in the single-objective optimization algorithms. In our method, the inter-task and intra-task learning are treated as two separated but cooperative procedures. DNL-MFPSO forms two kinds of neighbor structures by two distinct strategies.
2) MFCGA uses the consistent reproduction procedure in the whole searching. However, in our method, the particles can choose their inter-task transfer strategies to promote their search potentials.

Algorithm 3: Basic Structure of the Proposed Method.

1: Generate a swarm P of N_p particles by initializing their positions x_i ($i = 1, \ldots, N_p$);
2: Assign particles to a arbitrary task;
3: Perform evaluation of all particles in P;
4: Initialize the velocity v_i of each particle;
5: Assign the particles to the tasks so that every task occupies the same number of particles;
6: Store the task id of particle i in τ_i;
7: Initialize the personal best p_i of each particle;
8: Record the current best solution $g(t)$ of each task t;
9: Initialize the neighborhoods of particles via **Algorithm 1**;
10: $k = 1$
11: **Repeat**
12:   **For** $t = 1, \ldots, K$ **do**
13:     **If** task t has stagnated in G generations **then**
14:       Update the neighborhoods of each particle via **Algorithm 1**;
15:     **End If**
16:   **End For**
17:   Calculate the levels of particles via sorting the corresponding task objective values;
18:   **For each** $\boldsymbol{x}_i$ in P **do**
19:     Generate the exemplar for $\boldsymbol{x}_i$ via **Algorithm 2**;
20:     Update the velocities of $\boldsymbol{v}_i$ via Equation (9);
21:     Update the position $\boldsymbol{x}_i$ with velocity;
22:     Evaluate the individuals in P for selected optimization tasks only;
23:   **End For**
24:   Calculate the fitness of each particle in P via Equation (11);
25:   Update personal best position p of each particle according fitness;
26: Update the best positions $g(t), t = 1 \ldots K$;
27:   $k = k + 1$;
28: **Until**Reach the predefined maximum number of evaluations of function.

IV. EXPERIMENTS

In this section, we conduct comprehensive empirical experiments to evaluate the performance of DNL-MFPSO on both two-task and three-task benchmark optimization problems.

A. Test Problems

The first benchmark suit has three groups and total 9 benchmark problems with various properties, denoted by S1-S10. Each of them is a combination of the continuous functions from [29] with different dimensions and degrees of solution bias. The properties of three groups are described in Table II. The second benchmark suit includes the problems for CEC'17 MTO competition [48], denoted as S10-18. These benchmark problems are classified into three groups denoted as complete intersection (CI), partial intersection (PI) and no intersection (NI) according to the intersection of search spaces. Each category also could be further separated into three clusters based on the similarity of tasks, i.e. high similarity (HS), middle similarity (MS) and low similarity (LS), which infer different inter-task synergy. The third benchmark suit contains six problems E1-E3 and E7-E9 from the WCCI2020's benchmark problems consisted of more difficult tasks with non-separable decision space and multimodal function landscapes. The detailed specifications of this benchmarks are described in Section SI in supplementary materials. All benchmark functions can be found in website.[1]

[1] codes available in [Online]. Available: http://www.bdsc.site/websites/MTO/index.html

TABLE III
COMPARED ALGORITHMS

PSO Based Method	DE Based Method	DE Based Method
DNL-MFPSO MFPSO [18] MFPSO-ITL [20] DMSPSO [32] [49]	MFEA [26] MFEA-II [26] SBGA [26]	MFDE [18]

TABLE IV
PARAMETER SETTINGS

Parameter	Value
Population size (N_p)	100
Maximum NFEs (MAX_{NFE})	2×10^5
MAX_{NFE} for three tasks	5×10^5
Random mating parameter (rlp)	0.1
PSO Based Methods	
Maximum velocity (V_{max})	0.1
Local Acc. Coeff. (c_1)	2.05
Social Acc. Coeff. (c_2)	2.05
Inter-task Acc. Coeff. (MFPSO) (c_3)	0.8 [18]
Reforming Period (Ours) (G)	10
Neighbor Size (Ours) (N_t)	$20\%N_p$
Neighbor size (DMSPSO) (N_t)	$20\%N_p$ [32]
GA Based Methods	
Mutation probability (p_m)	0.1
Crossover probability (p_c)	0.8
DE Based Method	
Crossover probability (C_r)	0.9
The scale factor (F)	0.5
DE mutation strategy	DE/rand/1

Further, a set of multi-task neuro-evolution learning problems [4] are employed to testify the efficiency of the proposed algorithm on real-world applications.

B. Experimental Setup

We include three categories of compared algorithms in the literature. All algorithms used in experiments are shown in Table III. The parameters of the compared algorithms are set by referring to the citations. All parameter settings for these algorithms are listed in Table IV.

For all conducted experiments, the best, mean and standard deviation of each objective function is reported herein. The above measures are recorded in 30 independent runs when the number of evaluation calls reaches MAX_{NFE}. The results and corresponding analyses are presented as follows, while the results on three-task problems and the entire results of the compared algorithms on all 24 benchmarks are illustrated in Section SII and Table SVI in supplementary materials.

C. Observations of DNL-MFPSO

1) Exploration and Exploitation Investigation in DNL-MFPSO: In MFO, although utilizing the search experience of solving other related tasks can benefit the search performance, the excessive diversity could occur during the late stages and break the refined search. To verify DNL-MFPSO can reserve good inter-task exploration and exploitation and compromise two properties properly, we conduct an empirical experiment among DNL-MFPSO and MFPSO on two benchmarks (CIHS and unimodal–S10, NILS and multimodal–S18) with 50 dimensions in terms of the averaged diversity measure over two component tasks and global best objective function values.

Fig. 6 shows the experimental results of two compared algorithms on S10 and S18. The following observations are obtained. First, for tasks with completely intersected optimal solutions, inter-task exploitation should be more slightly emphasized, such that fast convergence can be achieved. From the diversity curves of each level and the convergence trends depicted in the first row of Fig. 6, it is observed that on S10 two algorithms bias exploitation, so that the diversity is decreased fast and the objective function values converge fast.

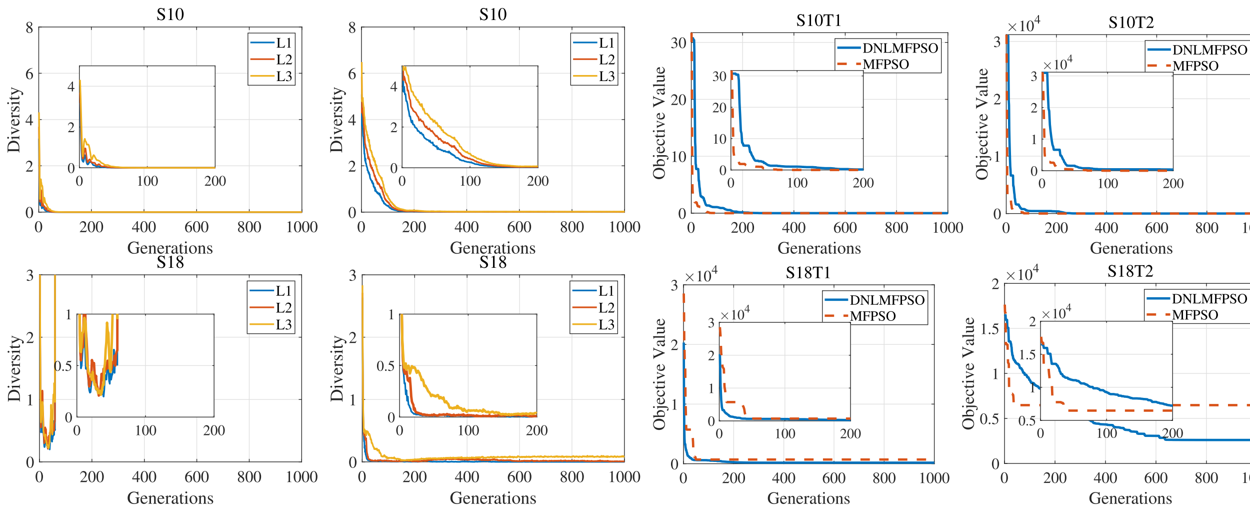
Second, for multimodal tasks with partially intersected optimal solutions, the inter-task transfer and refined search should be properly compromised without serious loss of refined search. As illustrated in the second row of Fig. 6, on S18, DNL-MFPSO achieves a good convergence speed and good quality of solutions, because the level-based learning strategy can help the particles in top level refine the promised areas and encourage the particles in lower levels to absorb the experience from foreign tasks. Fig. 6(a) shows that in MFPSO, the excessive perturbation occurs after a period of the fast convergence, such that the solutions are less improved during the later stages. This leads to poor performance of MFPSO in some functions.

Moreover, from the curves of diversity shown in Fig. 6(a)(b), the NDL-MFPSO can manage the diversity of swarm by the dynamic intra-task and inter-task local structure, such that the swarm can reserve the diversity in a relatively long period. Moreover, the level-based inter-task learning strategy makes use of the ability of the particles in different levels. Such that, particles in L1 accounting for refined search might have less diversity, while ones in L3 reserve more diversity.

2) Influences of Dynamic Local Topology and Level-Based Learning: In this part of experiments, we investigate the influence of three transfer operations, the level-based learning strategy and the local dynamic topology scheme on the proposed DNL-MFPSO. First, the proposed DNL-MFPSOs without the level-based learning strategy are denoted by appending a tag "Individual Migration (IM)," "interMediate Point (MP)" or "Differential Mutation (DM)" which represents the exemplar generation strategy. For example, MFPSO-IM represents the DNL-MFPSO with individual migration. MFPSO-MP indicates that the DNL-MFPSO uses intermediate point exemplar. MFPSO-DM represents the DNL-MFPSO with differential mutation. GMFPSO represents the MFPSO with the global social topology and level-based learning strategy.

Table V reports the experimental results with respect to objective function values among these versions. From the perspective of the local dynamic topology, we can find that from Table V on most problems the performance of the proposed DNL-MFPSO is comparable to that of the GMFPSO. The superiority of the local dynamic topology scheme is evident, especially in the comparison of DNL-MFPSO and GMFPSO on problems S12, S15 and S18, which have distant optima. In a word, we can see that the local dynamic topology strategy benefits the proposed DNL-MFPSO, especially on the complicated and low-similar problems, like problems S17, S18. This is because the local dynamic topology can reduce the converging speed of the PSO to preserve the diversity in the swarm. On the one hand, the local topology can retain the optima of multiple tasks. On the other

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) Swarm diversity in MFPSO (b) Swarm diversity in DNL-MFPSO (c)Objective value on T1 (d)Objective value on T2

Fig. 6. Swarm diversity and the global best fitness value comparison among DNL-MFPSO and GMFPSO on two benchmark problems 10 and 18.

TABLE V
DETAIL RESULTS OF THE EFFICIENCY OF COMPONENTS

Algorithm		DNL-MFPSO		GMFPSO(level-based)		DNL-MFPSO-IM		DNL-MFPSO-MP		DNL-MFPSO-DM	
Task		T1	T2	T1	T2	T1	T2	T1	T2	T1	T2
S10	min	0.0000E+00	0.0000E+00	0.0000E+00	1.3678E-13	0.0000E+00	3.9080E-14	2.2204E-16	8.8285E-13	2.2204E-16	1.5898E-12
	mean	0.0000E+00	0.0000E+00	9.1947E-03	1.6719E+02	6.4829E-03	1.1366E+02	6.7309E-03	1.6170E+02	8.2121E-03	1.8276E+02
	std	0.0000E+00	0.0000E+00	9.3892E-03	1.1014E+02	7.7626E-03	1.0738E+02	8.7094E-03	7.8063E+01	7.1345E-03	1.0404E+02
S11	min	1.5099E-14	0.0000E+00	2.4786E+00	7.8602E+01	2.8142E+00	7.0642E+01	2.8140E+00	1.0945E+02	3.6070E+00	8.2581E+01
	mean	9.0945E-01	1.0348E+01	5.1307E+00	2.4164E+02	5.1057E+00	2.3925E+02	4.6704E+00	2.5036E+02	5.6714E+00	2.8094E+02
	std	6.1872E-01	8.9239E+00	1.4395E+00	1.0251E+02	1.2293E+00	9.6432E+01	8.2158E-01	6.7868E+01	1.2354E+00	1.0134E+02
S12	min	8.7905E-01	2.3475E+03	5.8340E+00	4.6694E+03	1.0635E+01	2.7575E+03	4.7331E+00	4.8529E+03	6.7438E+00	4.3959E+03
	mean	2.0420E+00	3.4416E+03	1.9583E+01	6.8913E+03	1.9643E+01	5.5120E+03	1.1577E+01	8.2388E+03	1.6726E+01	8.4622E+03
	std	5.7335E-01	3.3839E+02	3.1698E+00	1.1296E+03	2.3138E+00	9.7716E+02	6.2729E+00	2.3776E+03	4.9606E+00	2.5488E+03
S13	min	6.2682E+01	6.0585E-28	3.0545E+02	8.2233E-25	3.6415E+02	1.3490E-25	3.9997E+02	7.6619E-25	3.1739E+02	2.2199E-23
	mean	1.1251E+02	4.5102E-27	5.4194E+02	3.2281E-08	6.9503E+02	6.4858E-17	5.8796E+02	2.8339E-05	7.0529E+02	1.4143E-04
	std	3.2797E+01	1.8393E-27	1.3433E+02	1.4792E-07	1.6103E+02	2.9881E-16	1.4003E+02	9.4473E-05	2.0189E+02	3.6184E-04
S14	min	7.9936E-15	1.0274E-02	1.2833E-03	2.7983E+01	6.1925E-09	1.9106E-02	4.9001E-12	2.4063E+01	4.2224E-02	4.2524E+00
	mean	1.5123E-01	4.4514E+01	2.5108E+00	8.3154E+01	2.2031E+00	7.0720E+01	2.4490E+00	7.2465E+01	2.4856E+00	8.4329E+01
	std	3.2648E-01	2.1369E+01	7.1196E-01	2.6012E+01	9.5719E-01	2.5544E+01	8.4962E-01	3.0097E+01	7.4598E-01	3.5167E+01
S15	min	3.9968E-14	0.0000E+00	7.4904E+00	7.2495E+00	9.4464E+00	9.6066E+00	7.8137E+00	6.2600E+00	5.9940E+00	6.9602E+00
	mean	1.7410E+00	3.3536E+00	1.6719E+01	1.6933E+01	1.7247E+01	1.7041E+01	1.4631E+01	1.4928E+01	1.6476E+01	1.6355E+01
	std	4.6514E-01	1.0704E+00	4.1228E+00	5.2064E+00	3.4785E+00	4.3542E+00	5.1078E+00	4.6252E+00	4.8329E+00	4.9245E+00
S16	min	7.6474E+00	0.0000E+00	6.3895E+00	9.9496E-01	4.0809E+00	4.1788E+01	5.8353E+00	7.6612E+01	8.6192E+00	4.6763E+01
	mean	3.1135E+01	1.1608E+00	5.8812E+01	1.7645E+02	5.2230E+01	2.2491E+02	6.5246E+01	2.1684E+02	1.6297E+02	2.3229E+02
	std	7.9757E+00	3.2706E+00	3.2530E+01	9.0926E+01	3.4251E+01	9.4317E+01	3.0222E+01	8.0963E+01	3.8537E+02	1.4216E+02
S17	min	0.0000E+00	8.1236E+00	5.7820E-13	1.9752E+01	4.4409E-15	2.0930E+01	4.1633E-14	1.9091E+01	1.1535E-13	1.9098E+01
	mean	4.3299E-16	1.1969E+01	6.7134E-03	2.5949E+01	4.3488E-03	2.7406E+01	7.0653E-03	2.4468E+01	7.0552E-03	2.8001E+01
	std	3.1837E-16	1.4484E+00	7.3191E-03	2.8085E+00	7.2870E-03	3.6766E+00	7.5528E-03	3.1866E+00	7.6290E-03	6.3706E+00
S18	min	6.2682E+01	2.5160E+03	3.9400E+02	2.4513E+03	4.2584E+02	4.0330E+03	3.6218E+02	4.4981E+03	3.3729E+02	4.2683E+03
	mean	1.1362E+02	3.4131E+03	8.2911E+02	6.9274E+03	8.8930E+02	6.1048E+03	6.7565E+02	8.8720E+03	7.5904E+02	8.4454E+03
	std	2.5804E+01	3.9566E+02	3.6153E+02	1.6233E+03	2.2460E+02	1.4637E+03	1.7552E+02	2.1847E+03	2.3268E+02	2.6264E+03
mean w/l		10/1	11/0	0/11	0/11	0/11	0/11	1/10	0/11	0/11	0/11

hand, each task can utilize the knowledge of other tasks in broad areas efficiently.

The performance of proposed DNL-MFPSO is obviously superior compared with the other versions of DNL-MFPSO on almost all problems. It is evident that DNL-MFPSO-IM has good performance on the problems with the similar optima, such as S10, S13, S16. Meanwhile, the DNL-MFPSO-MP and DNL-MFPSO-DM, which can explore the broad areas, is capable to achieve a good performance in the complicated problems, such as S17 and S18. Note that versions of DNL-MFPSO without level-based learning strategy also outperform the GMFPSO. This is because that the DE mutation operation can not provide enough shifting to the particles while the diversity of GMFPSO reduces rapidly. The verified usefulness of the level-based learning strategy can be ascribed to the trade-offs of the refining and the broad exploration. The individual migration and median point exemplar can potentially exploit the good positions of the particles. The shifting provided by the DE can afford more chances for MFPSO to escape from local optimal areas.

In addition, we further investigate the influence of the local dynamic topology scheme and the level-based learning strategy on the converging trends of the proposed MFPSO. The converging curves of all tasks are plotted in Fig. 7 in the supplementary material. We show some plots in Fig. 7. From the figure, we can observe that the DNL-MFPSO can obtain a significant converging speed on the relatively simple problems, such as

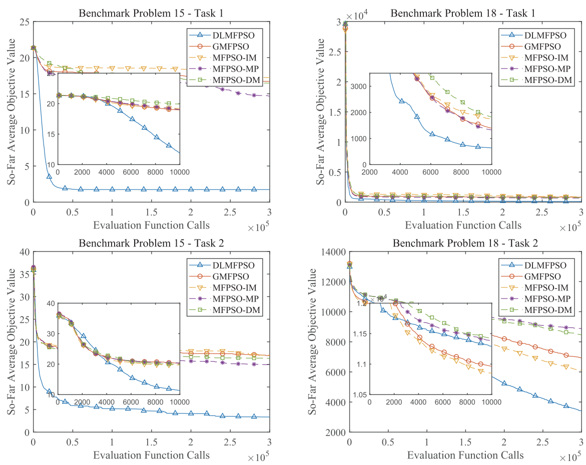


Fig. 7. Converagence curves of DNL-MFPSO, GMFPSO, MFPSO-IM, MFPSO-MP and MFPSO-DM on four benchmark problems 15 and 18.

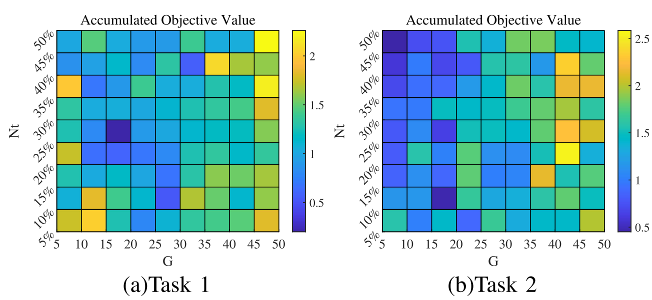


Fig. 8. Joint sensitivity investigation of N_t and G. The color patches show the accumulated normalized objective errors over S10, S13, S15. The objective values are normalized into [0,1] uniformly.

S8. This is ascribed to the level-based learning strategy, which can retain the good solutions in the first level. Meanwhile, the DNL-MFPSO can also keep the high diversity in the swarm by utilizing the local dynamic topology and the DE mutation operations in the complicated problems, such as S12, S15 and S18. It is observed that at the beginning of the DNL-MFPSO can control the converging speed. So the proposed DNL-MFPSO can obtain the better solutions than the other versions of DNL-MFPSO can do.

3) Sensitivity Analysis of Parameters: To analyze the influences of the parameters, we carry out the sensitivity analysis on the neighborhood size N_t and the period of the local dynamic strategy G in this experiment. We perform the proposed DNL-MFPSO with $N_t \in [5\% \times N, 10\% \times N, \ldots, 50\% \times N]$, where $N = N_p/K$, along with $G \in [5, 50]$. 10×10 prefab settings $s = (N_t, G)$ of two parameters are considered in $\mathcal{S}$. For clarity, the normalized objective error ϵ is recorded. ϵ_s is given by $\epsilon_s = (f_s - f_{\min})/(f_{\max} - f_{\min})$, where f_s is the best objective values found at the parameter setting $s \in \mathcal{S}$; $f_{\max}$ and $f_{\min}$ are respectively the maximizer and minimizer among all $f_{s\in\mathcal{S}}$. The accumulated error is the sum of related objective errors on all benchmark problems. We also report the accumulated error to showcase the overall performance.

We plot the sum of normalized objective errors on three benchmarks S10, S13 and S15 under 10x10 prefab settings upon the 10x10 grid. Fig. 8 shows the accumulated objective errors in different parameter settings. When G falls in $[10, 25]$ and N_t in $[20\%N, 25\%N]$, the algorithm can obtain a good overall performance on Task 1 in S10, S13 and S15, as shown in Fig. 8(a). When G lies in $[5, 25]$ and N_t lies in $[15\%N, 50\%N]$, the algorithm gives good solutions on Task 2 as depicted in Fig. 8(b). Accordingly, $G = 15$ and $N_t = 25\% \times N$ are recommended in this paper.

D. Comparisons With State-of-The-Art Multifactorial Algorithms

In this part of experiments, we validate the superiority of the proposed DNL-MFPSO compared with several other state-of-the-art MFO algorithms and a baseline of single-task DMSPSO. The brief reviews of the algorithms can be referred in Section II-C.

The results are shown in Table VI. Especially, on the problems S10, S11, S14 and S16, DNL-MFPSO can achieve the best quality of solutions on both two tasks compared with other MFO algorithms. DNL-MFPSO can preserve inter-task transfer and refined search abilities, and particularly take the balance between two aspects properly to search the multiple search spaces. Such a powerful ability benefits from the combination of the proposed dynamic neighborhood structure and the level-based inter-task learning strategy, which offers the good information isolation and the methodically synthesis of the exemplar generations with diverse search biases.

The complete results of the compared algorithms on all 24 benchmarks are illustrated in Fig. SVI in supplementary materials. The second last row (*best/second*) of Fig. SVI summarizes the overall performance of the proposed DNL-MFPSO. The last row "$w/t/l$" gives the comparison results of DNL-MFPSO against the existing algorithms, where w, t and l represent the number of problems the DNL-MFPSO wins, ties and loses another algorithm, respectively. Although, DNL-MFPSO is not superior in all problems, it can obtain the competitive results on 20 out of 24 benchmark problems in terms of the first task and 14 out of 24 problems in terms of the second task. When solving complex multi-task optimization, like E1-3 and E8-10, DNL-MFPSO also showcases the superior performance even compared with the latest MFO algorithms, like MFEA2 and SBGA.

E. Preliminary Application Study

In this experiment, we want to verify the efficiency of the proposed DNL-MFPSO in a simple reinforcement learning task. Following the study in [4], this experiment involves simultaneously solving multiple double cart-pole balancing problems. The double cart-pole balancing problems are generally regarded as a standard benchmark for artificial controlling systems [50]. These problems are to train a neural controller for balancing two poles of different lengths hinged on the cart. Herein, we only consider the Markovian case, where the velocity of cart is fed as input to the controller. This is, a neural controller has 6 inputs, namely the cart position, the cart velocity, the rotation angle of longer pole, the rotation angle of shorter pole, the angular velocity of longer pole and the angular velocity of shorter pole. All inputs are normalized such that they fall into the range of $[-1, 1]$. The

TABLE VI
COMPARISON RESULTS OF DNL-MFPSO, MFPSO, MFPSO-ITL, MFEA, MFDE, DMSPSO. WITHIN BRACKETS WILCOXON'S RANK-SUM TEST RESULTS ARE SHOWN WITH + (WIN), = (TIE), AND - (LOSE)

Prob.		DNL-MFPSO		MFPSO		MFPSO-ITL		MFEA2		MFEA		SBGA		MFDE		DMSPSO	
		T_1	T_2	T_1	T_2	T_1	T_2	T_1	T_2	T_1	T_2	T_1	T_2	T_1	T_2	T_1	T_2
S10	μ	**0.00e+00**	**0.00e+00**	8.23e-03(-)	1.02e+02(-)	6.73e-03(-)	1.62e+02(-)	9.07e-03(-)	5.03e+01(-)	2.55e-02(-)	1.49e+02(-)	4.72e-01(-)	1.47e+02(-)	6.36e-02(-)	*3.52e+01*(-)	*5.58e-03*(-)	5.78e+01(-)
	σ	0.00e+00	0.00e+00	1.12e-02	6.98e+01	8.71e-03	7.81e+01	7.60e-03	4.35e+01	9.76e-03	4.57e+01	5.04e-01	1.56e+02	4.28e-02	1.01e+01	6.75e-03	4.31e+01
S11	μ	**9.09e-01**	**1.03e+01**	3.03e+00(-)	1.31e+02(-)	4.67e+00(-)	2.50e+02(-)	2.05e+00(-)	7.08e+01(-)	3.59e+00(-)	1.80e+02(-)	1.32e+01(-)	1.50e+03(-)	*1.48e+00*(-)	*2.81e+01*(-)	2.71e+00(-)	7.09e+01(-)
	σ	6.19e-01	8.92e+00	5.50e-01	8.25e+01	8.22e-01	6.79e+01	9.89e-02	1.68e+01	6.32e-01	6.33e+01	1.92e+00	5.71e+02	3.94e-01	7.95e+00	6.44e-01	6.24e+01
S12	μ	*2.04e+00*	3.44e+03	2.83e+00(-)	*1.22e+03*(+)	1.16e+01(-)	8.24e+03(-)	**1.71e+00**(=)	1.89e+03(+)	1.93e+01(-)	2.57e+03(+)	2.00e+01(-)	**2.51e+02**(+)	7.86e+00(-)	1.46e+03(+)	2.83e+00(-)	1.01e+04(-)
	σ	5.73e-01	3.38e+02	6.25e-01	3.29e+02	6.27e+00	2.38e+03	3.01e-01	3.21e+02	3.03e+00	2.96e+02	3.93e-01	1.51e+02	2.01e+00	9.56e+02	7.22e-01	1.68e+03
S13	μ	*1.13e+02*	**4.51e-27**	2.15e+02(-)	1.00e-02(-)	5.88e+02(-)	2.83e-05(-)	**1.11e+02**(=)	3.25e-03(-)	4.81e+02(-)	3.84e-02(-)	1.49e+03(-)	5.01e+01(-)	1.96e+02(-)	2.20e+01(-)	2.37e+02(-)	*1.13e-08*(-)
	σ	3.28e+01	1.84e-27	4.76e+01	5.39e-02	1.40e+02	9.45e-05	2.25e+01	1.17e-03	8.58e+01	1.39e-02	3.66e+02	7.59e+01	4.16e+01	1.44e+01	6.15e+01	1.82e-08
S14	μ	**1.51e-01**	**4.45e+01**	2.08e+00(-)	1.02e+02(-)	2.45e+00(-)	7.25e+01(-)	*1.47e+00*(-)	1.01e+02(-)	2.66e+00(-)	1.62e+02(-)	6.12e+00(-)	1.78e+04(-)	1.63e+00(-)	1.73e+02(-)	2.82e+00(-)	*5.08e+01*(-)
	σ	3.26e-01	2.14e+01	8.03e-01	3.04e+01	8.50e-01	3.01e+01	7.74e-01	5.54e+01	5.38e-01	3.43e+01	3.43e+00	3.32e+04	5.61e-01	6.43e+01	7.17e-01	1.72e+01
S15	μ	**1.74e+00**	3.35e+00	2.79e+00(-)	2.13e+00(+)	1.46e+01(-)	1.49e+01(-)	*1.91e+00*(=)	2.35e+00(+)	1.99e+01(-)	1.70e+01(-)	2.01e+01(-)	2.51e+01(-)	4.52e+00(-)	*9.81e-01*(+)	2.94e+00(-)	**5.57e-01**(+)
	σ	4.65e-01	1.07e+00	7.82e-01	6.54e-01	5.11e+00	4.63e+00	3.86e-01	1.48e-01	1.11e-01	2.78e+00	4.11e-01	2.32e+00	8.65e-01	4.82e-01	7.18e-01	7.47e-01
S16	μ	**3.11e+01**	**1.16e+00**	1.25e+02(-)	1.33e+02(-)	*6.52e+01*(-)	2.17e+02(-)	1.35e+02(-)	1.29e+02(-)	1.59e+02(-)	2.02e+02(-)	9.75e+03(-)	3.24e+02(-)	2.76e+02(-)	*4.89e+01*(-)	8.27e+01(-)	8.27e+01(-)
	σ	7.98e+00	3.27e+00	1.29e+02	9.34e+01	3.02e+01	8.10e+01	2.71e+01	2.14e+01	4.42e+01	7.67e+01	2.60e+04	3.65e+02	1.18e+02	1.61e+01	3.53e+01	7.72e+01
S17	μ	**4.33e-16**	1.20e+01	6.62e-03(-)	1.46e+01(-)	7.07e-03(-)	2.45e+01(-)	*4.27e-03*(-)	1.93e+01(-)	3.26e-02(-)	2.43e+01(-)	4.80e-01(-)	3.12e+01(-)	3.84e-01(-)	*9.66e+00*(+)	4.84e-03(-)	**4.26e+00**(+)
	σ	3.18e-16	1.45e+00	7.89e-03	2.67e+00	7.55e-03	3.19e+00	2.74e-03	1.83e+00	1.23e-02	2.92e+00	5.12e-01	1.94e+00	1.99e-01	1.74e+00	6.62e-03	1.69e+00
S18	μ	*1.14e+02*	3.41e+03	2.14e+02(-)	*1.25e+03*(+)	6.76e+02(-)	8.87e+03(-)	**1.11e+02**(=)	1.90e+03(+)	5.02e+02(-)	2.72e+03(+)	1.49e+03(-)	**3.47e+02**(+)	5.02e+02(-)	1.41e+03(+)	2.39e+02(-)	1.03e+04(-)
	σ	2.58e+01	3.96e+02	3.95e+01	3.53e+02	1.76e+02	2.18e+03	1.36e+01	5.97e+02	8.46e+01	5.42e+02	2.64e+02	2.05e+02	1.18e+02	8.24e+02	6.77e+01	1.04e+03

TABLE VII
RESULTS OF TWO-TASK DOUBLE CART-POLE BALANCING PROBLEMS

Task	T_1, T_2	T_1, T_3	T_2, T_3
l_s(m)	0.60, 0.65	0.60, 0.70	0.65, 0.70
DNL-MFPSO	37%, 27%	27%, 13%	33%, 27%
MFPSO	23%, 23%	17%, 5%	10%, 5%
MFPSO-ITL	27%, 27%	17%, 3%	13%, 7%
MFEA-II	30%, 27%	30%, 7%	27%, 27%
MFEA	37%, 27%	47%, 7%	10%, 7%
MFDE	57%, 47%	40%, 10%	17%, 0%
DMSPSO	17%, 0%	13%, 0%	0%, 0%

other parameter settings and the environment settings refer to the same settings in [4] and [51]. The length of the longer pole is fixed in 1.0 m, while that of shorter one varies in 0.6 m, 0.65 m, 0.7 m. When the length of shorter pole gets closer to 1.0 m, the balancing task gets harder [50]. The results of the proposed method on the neuro-evolution learning problems are illustrated in Table VII.

According to the results in Table VII, it is shown that the inter-task transfer can help MFO methods (DNL-MFPSO, MFPSO, MFPSO-ITL, MFEA-II, MFEA and MFDE) seek a good controller for the balancing tasks at a higher success rate compared with the single-task PSO which successes at a rate of 17% on T_1, 0% on T_2, 0% on T_3. DNL-MFPSO can outperform MFPSO improving by up to 130% in terms of success rate. It also outperforms the other MFPSOs (MFPSO, MFPSO-ITL). Especially, on harder problems, the proposed method can conduct a favorable transfer by the proposed learning strategy. Thus, the proposed method can solve T_3 by the higher chance of 15% in (T_1, T_3) compared to the other MFOs. It also achieves an improvement on (T_2, T_3) with (33%,27%) success rates better than MFEA-II.

V. CONCLUSION

Taking advantage of the local topology PSO in preserving high diversity, we propose a multifactorial PSO algorithm with dynamic inter-task neighbors to tackle with multitask optimization. First, combined with the local dynamic topology scheme, DNL-MFPSO incorporates a level-based inter-task learning strategy, which takes into account the different aims of the particles. Specifically, we divide the swarm into three levels in which the particles are assigned with different searching behaviors. Particles in the top level are focused on refining the search area, and the secondary level of particles aims at exploring the broad areas. Subsequently, to realize the different tradeoffs of the exploitation and exploration, we employ different learning operations in different levels. Together, these techniques draw a good balance between exploration and refining, leading to competitive efficiency and effectiveness of the proposed DNL-MFPSO. In the experiments, efficiency of DNL-MFPSO is testified via comprehensive experiments.

Although, DNL-MFPSO demonstrates its potential in solving the multitask optimization, it still can not avoid negative transfer in complicated problems, leading to inferior performance in these problems. Therefore, there is much room to further improve the proposed algorithm's performance through an efficient transfer strategy. Moreover, the proposed method suffers a decreasing performance when the tasks increases, since it is hard to form a stable connection between the tasks having synergy by the simple random selection of source task in the inter-task learning process. Thus, we are eager to develop an adaptive way to select the source task.

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, Apr. 2016.
[2] A. H. Wright, M. D. Vose, and J. E. Rowe, "Implicit parallelism," in *Proc. Genetic Evolut. Comput. Conf.*, 2003, pp. 1505–1517.
[3] T. Back, U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 3–17, Apr. 1997.
[4] K. K. Bali, Y. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," *IEEE Trans. Evolut. Comput*., vol. 24, no. 1, pp. 69–83, Feb. 2020.
[5] Y. S. Ong, "Towards evolutionary multitasking: A new paradigm in evolutionary computation," in *Proc. Comput. Intell. Cyber Secur. Comput. Models*, Singapore, 2015, pp. 25–26.

[6] 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, Mar. 1979.

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

[8] 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 *Proc. IEEE Reg. 10 Conf.*, Singapore, 2016, pp. 3157–3164.

[9] R. Sagarna and Y. S. Ong, "Concurrently searching branches in software tests generation through multitask evolution," in *Proc IEEE Symp. Ser. Comput. Intell*, Athens, Greece, 2016, pp. 1–8.

[10] 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 *Proc. IEEE Symp. Ser. Comput. Intell*, Singapore, 2016, pp. 1–8.

[11] R. Chandra, A. Gupta, Y. S. Ong, and C.-K. Goh, "Evolutionary multi-task learning for modular training of feedforward neural networks," in *Neural Inf. Process*. Cham: Springer, 2016, pp. 37–46.

[12] C. A. Coello Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[13] W. N. Chen, J. Zhang, H. S. H. Chung, W. L. Zhong, W. G. Wu, and Y. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, Oct. 2010.

[14] X. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, Oct. 2011.

[15] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Trans. Syst.*, *Man, Cybern. B., Cybern.*, vol. 42, no. 3, Nov. 2012.

[16] H. Duan, Q. Luo, Y. Shi, and G. Ma, "Hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration," *IEEE Comput. Intell. Mag.*, vol. 8, no. 3, pp. 16–27, Jul. 2013.

[17] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 1–18, Dec. 2015.

[18] L. Feng *et al.*, "An empirical study of multifactorial PSO and multifactorial DE," in *Proc. IEEE Congr. Evol. Comput., Rio*, 2017, pp. 921–928.

[19] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Trans. Cyber.*, vol. 46, no. 10, pp. 2238–2251, Sep. 2016.

[20] B. Zhang, A. K. Qin, and T. Sellis, "Evolutionary feature subspaces generation for ensemble classification," in *Proc. Gen. Evol. Comput. Conf.*, Kyoto, 2018, pp. 577–584.

[21] C. Yue, B. Qu, and J. Liang, "A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems," *IEEE Trans. Evolut. Comput.*, vol. 22, no. 5, pp. 805–817, Oct. 2018.

[22] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang, "An adaptive multipopulation framework for locating and tracking multiple optima," *IEEE Trans. Evolut. Comput.*, vol. 20, no. 4, pp. 590–605, Aug. 2016.

[23] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Trans. Evolut. Comput.*, vol. 19, no. 2, pp. 246–263, Apr. 2015.

[24] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010.

[25] T. O'Brien and D. Guiney, *Differentiation in Teaching and Learning: Principles and Practice*. Wiley Online Library, 2001.

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

[27] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. Int. Symp. Micro Mach. Human Sci.*, Nagoya, Japan, 1995, vol. 1, pp. 39–43.

[28] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[29] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, May 2006.

[30] S. L. Sabat, L. Ali, and S. K. Udgata, "Integrated learning particle swarm optimizer for global optimization," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 574–584, Jan. 2011.

[31] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Trans. Cyber.*, vol. 46, no. 10, pp. 2238–2251, Sep. 2016.

[32] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism," in *Proc. IEEE Congr*. Evol. Comput., 2006, pp. 9–16.

[33] Q. Yang, W. Chen, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 578–594, Sep. 2018.

[34] 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., San Sebastian, Spain, 2017, pp. 1295–1302.

[35] Y. W. Wen and C. K. Ting, "Parting ways and reallocating resources in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput.*, San Sebastian, Spain, 2017, pp. 2404–2411.

[36] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W.-H. Tsang, "Memetic search with interdomain learning: A realization between CVRP and CARP," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 644–658, Oct. 2015.

[37] Y. W. Wen and C. K. Ting, "Learning ensemble of decision trees through multifactorial genetic programming," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, Canada, 2016, pp. 5293–5300.

[38] R. Liaw and C. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *Proc. IEEE Conf. Evol. Comput.*, San Sebastian, Spain, 2017, pp. 2266–2273.

[39] R.-T. Liaw and C.-K. Ting, "Evolutionary manytasking optimization based on symbiosis in biocoenosis," in *Proc. AAAI Conf. Art. Intel.*, Honolulu, USA, 2019, pp. 4295–4303.

[40] Y. Chen, J. Zhong, L. Feng, and J. Zhang, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 3, pp. 369–384, Jun. 2020.

[41] D. Liu, S. Huang, and J. Zhong, "Surrogate-assisted multi-tasking memetic algorithm," in *Proc. IEEE Conf. Evol. Comput.*, Rio de Janeiro, Brazil, 2018, pp. 1–8.

[42] Y. Chen, J. Zhong, and M. Tan, "A fast memetic multi-objective differential evolution for multi-tasking optimization," in *Proc. IEEE Conf. Evol. Comput.*, Rio de Janeiro, Brazil, 2018, pp. 1–8.

[43] S. Huang, J. Zhong, and W. Yu, "Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization," *IEEE Trans. Emerg. Top. Comput*., 2019. [Online]. DOI:10.1109/TETC.2019.2945775.

[44] L. Feng *et al.*, "Evolutionary multitasking via explicit autoencoding," *IEEE Trans. Cyber*., vol. 49, no. 9, pp. 3457–3470, Sep. 2019.

[45] J. Tang, Y. Chen, Z. Deng, Y. Xiang, and C. P. Joy, "A group-based approach to improve multifactorial evolutionary algorithm," in *Proc. Interna. Joint Conf. Art. Intell.*, Stockholm, Sweden, 2018, pp. 3870–3876.

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

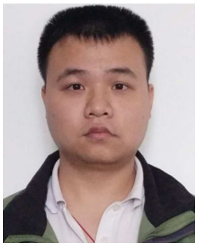
[47] E. Osaba, A. D. Martinez, J. L. Lobo, J. D. Ser, and F. Herrera, "Multifactorial cellular genetic algorithm (MFCGA): Algorithmic design, performance comparison and genetic transferability analysis," in *Proc. IEEE Congr. Evol. Comput.*, Glasgow, U.K., 2020, pp. 1–8.

[48] B. Da *et al.*, "Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metrics and baseline results," 2020, *arXiv:1706.03470*.

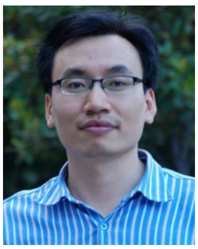
[49] S. Jagannath and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm Evol. Comput.*, vol. 16, pp. 1–18, Jun. 2014.

[50] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Accelerated neural evolution through cooperatively coevolved synapses," *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, May 2008.

[51] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, Mar. 2002.



Zedong Tang received the B.Eng. and Ph.D. degrees from Xidian University, Xi'an, China, in 2014 and 2020, respectively. Since 2020, he has been a Lecturer with the Academy of Advanced Interdisciplinary Research, Xidian University. His current research interests include computational intelligence and machine learning.



Maoguo Gong (Senior Member, IEEE) received the B.S. degree in electronic engineering (Hons.) and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively. Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted to an Associate Professor and as a Full Professor, both with exceptive admission. His research interests include computational intelligence with applications to optimization, learning, data mining, and image understanding. Dr. Gong was the recipient of the prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Yu Xie received the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2020. He is currently a Teacher with the School of Computer and Information Technology, Shanxi University, Taiyuan, China. He has authored or coauthored more than 30 articles on his topic of interests in international journals. His research interests include deep representation learning, graph neural networks, and artificial intelligence safety.

Hao Li received the B.S. degree in electronic engineering and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2013 and 2018, respectively. He is currently an Associate Professor with the School of Electronic Engineering, Xidian University. His research interests include computational intelligence and machine learning.

A. K. Qin (Senior Member, IEEE) received the B.Eng. degree from Southeast University, Nanjing, China, in 2001 and the Ph.D. degree from Nanyang Technology University, Singapore, in 2007. From 2007 to 2017, he was with the University of Waterloo, Waterloo, ON, Canada, INRIA, Grenoble-Rhône-Alpes, France, and RMIT University, Melbourne, VIC, Australia. In 2017, he joined an Associate Professor with the Swinburne University of Technology, Melbourne, VIC, Australia, where he is currently the Director of Swinburne Intelligent Data Analytics Laboratory, the Program Lead of Swinburne Data Science Research Institute, and the Leader of Machine Learning and Intelligent Optimisation (MLIO) Research Group. His research interests mainly include machine learning, evolutionary computation, computer vision, remote sensing, services computing, and pervasive computing. Dr. Qin was the recipient of the 2012 IEEE Transactions on Evolutionary Computation Outstanding Paper Award and the Overall Best Paper Award at the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems in 2014. He is currently the Vice-Chair of the IEEE Neural Networks Technical Committee and the IEEE Emergent Technologies Task Forces on Collaborative Learning and Optimisation and Multitask Learning and Multitask Optimisation.