Multi-Swarm Collaborative Particle Swarm Optimization Algorithm with Comprehensive Dimension Learning (Postprint)

Authors: Zhang Qiwen, Yangting Wang

Date: 2022-04-07T00:00:00+00:00

Abstract

To address the "over-exploitation" issue in the Dimensional Learning Strategy (DLS), a Comprehensive Dimensional Learning Multi-swarm Cooperative Particle Swarm Optimization algorithm (CDL-MCPSO) is proposed. To improve population search efficiency, the algorithm employs a cluster structure based on the master-slave paradigm, dividing the population into one master swarm and four slave swarms. The master swarm executes a comprehensive learning strategy to conduct large-scale exploration in the search space, while the slave swarms execute the Comprehensive Dimensional Learning (CDL) strategy to perform high-precision exploitation near local optima. The master and slave swarms, by executing algorithms with different functions, can effectively achieve a balance between exploration and exploitation. Meanwhile, to maintain population diversity, a novel Solution Exchange Mechanism (SEM) is proposed to facilitate information exchange and cooperation after the master and slave swarms have independently run their respective algorithms for several generations, thereby guiding particles to conduct more accurate searches in later stages. Finally, to address the excessive randomness in the initialization process, the Latin Hypercube Sampling method is employed to reconstruct the input distribution for the algorithm. To validate the effectiveness of CDL-MCPSO, it is compared with five particle swarm optimization algorithm variants on ten test functions; the results demonstrate that the algorithm can consistently find solutions superior to or comparable with those of the comparison algorithms, exhibiting feasibility and high efficiency in solving complex functions.

Full Text

Preamble

Vol. 39 No. 8
Application Research of Computers
ChinaXiv Cooperative Journal

Multi-Swarm Collaborative Particle Swarm Optimization Algorithm Based on Comprehensive Dimensional Learning

Zhang Qiwen, Wang Yangting†
(School of Computer & Communication, Lanzhou University of Technology, Lanzhou 730050, China)

Abstract: To address the "over-exploitation" problem in the Dimensional Learning Strategy (DLS), this paper proposes a Multi-Swarm Collaborative Particle Swarm Optimization algorithm with Comprehensive Dimensional Learning (CDL-MCPSO). To improve population search efficiency, the algorithm adopts a cluster structure based on the master-slave paradigm, dividing the population into one master group and four slave groups. The master group executes a comprehensive learning strategy to conduct large-scale exploration in the search space, while the slave groups execute the Comprehensive Dimensional Learning (CDL) strategy for high-precision exploitation near local optima. By executing algorithms with different functions, the master-slave groups can effectively achieve a balance between exploration and exploitation. To maintain population diversity, a new Solution Exchange Mechanism (SEM) is proposed to facilitate information exchange and cooperation after the master and slave groups have independently run their respective algorithms for several generations, thereby guiding particles toward more accurate searches in later stages. Finally, to address the excessive randomness in the initialization process, Latin Hypercube Sampling is employed to reconstruct the input distribution. To verify the effectiveness of CDL-MCPSO, it is compared with five PSO variants on ten test functions. Experimental results demonstrate that the algorithm consistently finds solutions that are better than or equivalent to those of the comparison algorithms, showing feasibility and efficiency in solving complex functions.

Keywords: Particle Swarm Optimization (PSO); Comprehensive Dimensional Learning Strategy; Master-Slave Paradigm; Multi-Group Collaboration

---

0 Introduction

In recent years, intelligent optimization algorithms have gradually become the preferred method for solving many complex optimization problems in scientific research and engineering applications. Common examples include Particle Swarm Optimization (PSO), Harmony Search (HS), Cultural Algorithm (CA), Teaching-Learning-Based Optimization (TLBO), and Gravitational Search Algorithm (GSA). The most classic among these is the Particle Swarm Optimization algorithm (PSO) [1], which simulates biological behavioral mechanisms and was formally proposed by Dr. Kennedy and Eberhart in 1995 as a population-based metaheuristic optimization algorithm. Due to its simple implementation and few control parameters, it has attracted extensive research and numerous variants. Improvements to PSO mainly focus on four aspects: parameter settings, topological structures, algorithm fusion, and learning strategies.

Parameter values play a crucial role in balancing the exploration and exploitation capabilities of algorithms. Shi et al. [2] discovered that a particle's inheritance ability from its previous velocity significantly impacts algorithm performance, leading to the introduction of inertia weight in the basic PSO algorithm. Sarhani et al. [3] proposed an inertia weight control mechanism based on Hidden Markov Model state estimation to balance global and local search capabilities, with experimental results demonstrating convergence advantages. In 2017, Gou et al. [4] dynamically adjusted parameter values according to individual differences among particles while employing a restart strategy to regenerate particles, proving the algorithm's robustness and scalability.

Regarding topological structure selection, Nasir et al. [5] proposed a Dynamic Neighborhood Learning Particle Swarm Optimization algorithm (DNLPSO) that selects sample particles from the best positions in the neighborhood (including itself). Reference [6] conducted experimental analysis on various topological structures to fundamentally overcome PSO's tendency to fall into local optima, ultimately proposing a hybrid topology PSO algorithm that integrates global exploration and local exploitation.

Utilizing different evolutionary algorithms to improve PSO performance is another research focus. The earliest algorithm fusion was by Gong et al. [7], who introduced three classic genetic operations—selection, mutation, and crossover—into PSO. Each particle generates a promising sample through genetic operations and then learns from the sample in the standard PSO manner. In 2018, Aydilek et al. [8] integrated concepts from the Firefly Algorithm and PSO, with experimental results proving that the algorithm not only effectively balances exploration and exploitation but also significantly improves runtime and solution accuracy. Senel et al. [9] combined PSO's exploitation capability with the exploration capability of Grey Wolf Optimization, achieving good results in a short time under the same test environment, demonstrating that the algorithm can find better optimization schemes in less time.

Based on these various improvements, it is clear that the main objectives of PSO variants are to maintain a balance between population exploration and exploitation capabilities, improve algorithm convergence speed, and enhance solution quality while preserving population diversity. However, most real-world optimization problems are complex. Regarding diversity, single-modal problems require smaller population diversity for rapid convergence, while multimodal problems require better population diversity to avoid falling into local extrema.

Thus, employing a single strategy cannot meet the needs of different optimization problems. Therefore, some researchers have proposed divide-and-conquer strategies to solve complex optimization problems, applying different learning strategies for different optimization objectives. Chen et al. [10] solved multi-objective problems based on comprehensive learning strategies, updating dominant particles through decomposition methods to enhance solution distribution, using an archive mechanism to store non-dominated solutions during optimization, and employing polynomial mutation to avoid local optima. Mendes et al. [11] proposed the Fully Informed Particle Swarm Optimization algorithm (FIPS), which updates particle positions using the average of all neighbors' personal best information in a fully connected neighborhood. Reference [12] proposed a multi-subgroup PSO algorithm with game probability selection after considering factors such as population structure, multi-modal learning, and inter-individual games. Parsopoulos et al. [13] proposed the Unified Particle Swarm Optimization algorithm (UPSO), which updates particle positions using the best information from global and local neighborhoods by selecting an appropriate neighborhood size. Yuan et al. [14] dynamically divided the population into three different classes and employed local learning models, standard learning models, and global learning models according to the characteristics of particles in different classes to improve algorithm performance. Reference [15] developed two new learning strategies: on one hand, proposing a local sparsity measurement method in the fitness landscape to estimate particle congestion and distribution, and establishing an exploration strategy by guiding particles to sparse areas; on the other hand, proposing an adaptive exploitation strategy based on multi-swarm strategy and adaptive subgroup size adjustment. Zhou et al. [16] proposed an adaptive hierarchical update PSO algorithm that generates two-layer and three-layer update formulas for two groups and introduced a multi-choice comprehensive learning strategy. Şaban et al. [17] proposed the Parallel Comprehensive Learning Particle Swarm Optimization algorithm (PCLPSO) based on comprehensive learning and the master-slave paradigm, which performs solution information cooperation between different paradigms. Reference [18], inspired by astronomy and botany, proposed a multi-strategy fusion PSO algorithm, where three-black-hole system capture strategy and multi-dimensional random disturbance strategy enhance global search capability, while coordination factors complete the transition from global to local search to improve convergence speed. Reference [19] proposed a dual-population comprehensive learning PSO algorithm based on particle replacement (PP-CLPSO) according to PSO's convergence characteristics and chaotic ideas from Logistic mapping.

From these improvement methods, it is evident that most modifications targeting parameters, topological structures, and algorithm fusion only optimize one or two objectives without balancing relationships among multiple objectives. The divide-and-conquer philosophy based on multiple strategies designs different learning strategies according to different optimization objectives of the algorithm, which can not only achieve a balance between exploration and exploitation but also effectively avoid premature convergence and improve convergence speed and solution quality.

CDL-MCPSO adopts the same divide-and-conquer philosophy as these multi-strategy fusion improvements but considers the "oscillation" and "dimension degradation" phenomena that most algorithms ignore. For example, the Dimensional Learning Strategy (DLS) proposed in reference [20] solved the "dimension degradation" phenomenon but created a new problem—"over-exploitation"—due to the learning strategy not being fully adapted to subgroup functions across different groups. The reason is that the second term of the formula uses the global best as the learning sample. After a few iterations, it becomes identical to the third term, both playing an exploitation role, doubling the exploitation capability and gradually losing population diversity. To address this deficiency, CDL-MCPSO proposes the "Comprehensive Dimensional Learning Strategy (CDL)" by integrating the advantages of Comprehensive Learning Strategy (CLS) and DLS. It removes the global best term from the third part of the formula and introduces the probabilistic learning idea from CLS for all other particles, allowing particles to explore appropriately while exploiting. Based on information flow principles, a new Solution Exchange Mechanism (SEM) is proposed to improve both convergence speed and solution quality. Finally, Latin Hypercube Sampling reconstructs the input distribution to effectively avoid negative impacts from excessive randomness in the initialization process. The description of the improvement rationale demonstrates that CDL-MCPSO possesses novelty and cutting-edge characteristics.

1.1 Particle Swarm Optimization Algorithm

PSO [2] originates from research on the social behavior of bird flocks sharing individual knowledge during foraging. Each individual in the swarm is called a particle, representing a potential solution in the search space. Particles update their velocity and position by following their own personal best and the swarm's global best until an optimal solution satisfying termination conditions is found. The particle velocity and position update formulas are as follows:

$$
\begin{aligned}
V_{i,j}(t+1) &= \omega V_{i,j}(t) + c_1 r_{1,j}(pbest_{i,j} - X_{i,j}(t)) + c_2 r_{2,j}(gbest_j - X_{i,j}(t)) \\
X_{i,j}(t+1) &= X_{i,j}(t) + V_{i,j}(t+1)
\end{aligned}
$$

where $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$ represent the population size and dimensionality, respectively. ω denotes the inertia weight, c_1 and c_2 are cognitive and social coefficients, and $r_{1,j}$ and $r_{2,j}$ are two random numbers uniformly distributed in [0,1] for the j-th dimension. $pbest_{i,j}$ and $gbest_j$ represent the particle's personal best and global best positions.

1.2 Comprehensive Learning Strategy (CLS)

In 2006, Liang et al. [21] proposed CLPSO, which employs a new learning strategy that updates velocity by utilizing all other particles' personal best information, providing particles with more learning samples and potential search space. In the comprehensive learning strategy, particle updates are guided by learning samples $X_{cl,i,j}(t)$. The velocity update formula is:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{i,j}(X_{cl,i,j}(t) - X_{i,j}(t))$$

Position updates follow equation (2), where $X_{cl,i,j}(t)$ represents the sample generated after particle i's personal best $pbest_{i,j}$ learns from all other particles' personal bests in dimension j (including particle i's own personal best in dimension j). The sample learning process works as follows: which particle's personal best each dimension of particle i learns from depends on the learning probability P_c. Different particles have different learning probabilities. First, a random number `rand` is generated for each dimension. If `rand` is greater than P_c, the particle learns from its own personal best in the corresponding dimension; otherwise, it learns from other particles' personal bests. The learning probability P_c update formula is:

$$P_c = 0.05 + 0.45 \times \frac{\exp(10 \times (i-1)/(ps-1)) - 1}{\exp(10) - 1}$$

where ps represents the population size. In the comprehensive learning strategy, other particles' previous best positions are samples that any particle can learn from, and each dimension of a particle can learn from different samples. This strategy of fusing information from different learning samples expands the particle's potential search space and greatly improves the population's global search capability, making it excellent for solving multimodal problems. However, due to the removal of the global best term, its performance on unimodal problems is suboptimal.

1.3 Dimensional Learning Strategy (DLS)

Xu et al. [20] proposed the Dimensional Learning Strategy to address CLPSO's shortcomings and the "oscillation" [22] and "two steps forward, one step back" [23] problems in standard PSO's population update. DLS updates velocity using a constructed learning sample $X_{dli,j}$ and the global best $gbest$. The learning sample $X_{dli,j}$ is constructed differently from CLS: while CLS's learning sample $X_{cl,i,j}$ each dimension comes from its own or all other particles' corresponding dimensions, DLS's learning sample $X_{dli,j}$ each dimension comes from its own or the global best $gbest$'s corresponding dimension, with learning depending on whether the particle's fitness improves after dimension replacement. This strategy can transmit excellent information from *gbest* to sample $X_{dli,j}$, ensuring the learned sample is no worse than the personal best. The velocity update formula is:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{i,j}(X_{dli,j} - X_{i,j}(t)) + c_2 r_{2,j}(gbest_j - X_{i,j}(t))$$

Position updates follow the standard PSO update equation (2). DLS not only solves the "oscillation" and "dimension degradation" phenomena but also greatly accelerates algorithm convergence by adding back the global best learning term. However, since the last two terms become similar after a few generations, both focusing on exploitation, "over-exploitation" easily occurs.

---

2 Comprehensive Dimensional Multi-Swarm Collaborative Particle Swarm Optimization Algorithm (CDL-MCPSO)

Existing multi-swarm learning strategies ignore the dimension degradation phenomenon, causing the "two steps forward, one step back" problem during particle position updates. Even when particle fitness improves, it is only because in some dimensions the particle approaches the global best more than it moves away in other dimensions, leading to particles oscillating between approaching and moving away from the global best, reducing search efficiency and affecting solution quality. Additionally, after analyzing the DLS velocity update formula, we found that the second term's constructed learning sample $X_{dli,j}$ gradually becomes very close to the global best *gbest* after several generations of learning. For subsequent particle updates, both the second and third terms' learning samples become the global best, doubling the exploitation capability of the last term due to the infinite proximity of the last two terms. Moreover, particles do not learn from other particles during the update process, lacking conditions to help them escape local optima, making "over-exploitation" highly likely. To solve this problem and improve algorithm efficiency, this paper first adopts a master-slave paradigm cluster structure: one master group and four slave groups. Second, the master and slave groups execute CLS with strong global exploration capability and the improved CDL strategy with strong exploitation capability, respectively. Finally, to address excessive initialization randomness, Latin Hypercube Sampling reconstructs the input distribution. With the dual strategies of multi-swarm and subgroup-function-adapted update rules, the algorithm's search efficiency is greatly improved. The proposed slave group update rule not only solves the "dimension degradation" problem but also avoids "over-exploitation." The CDL-MCPSO algorithm is detailed below.

2.1 Master-Slave Paradigm Cluster Structure

To improve the balance between exploration and exploitation, a cluster structure based on the master-slave paradigm is adopted, dividing the entire population into two types of groups: one master group ($Master$) and four slave groups ($Slave_i, i = 1, 2, 3, 4$). The master group executes the comprehensive learning strategy with strong global search capability to conduct large-scale exploration in the search space, while slave groups execute the improved strategy—Comprehensive Dimensional Learning (CDL)—for high-precision exploitation near already-found better solutions. The master-slave paradigm cluster structure is illustrated in [Figure 1: see original paper].

2.2 Master-Slave Group Update Strategies

To enable slave groups to conduct high-precision exploitation near local optima, the improved DLS (CDL) is selected as their update rule. The unimproved DLS was proven in reference [19] to have strong exploitation capability, but its "over-exploitation" problem can easily lead particles into local optima. The improvement removes the global best learning term from DLS's velocity update formula to prevent the sample $X_{dli,j}$ constructed after several generations of learning from infinitely approaching the global best $gbest$, and adds the second term from CLS where particles learn from all other particles' personal bests. This allows CDL to appropriately weaken DLS's exploitation level, preventing premature convergence and enabling DLS to achieve its best optimization effect. The velocity update formula is as follows:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_{i,j}(X_{cl,i,j} - X_{i,j}(t)) + c_2 r_{2,j}(X_{dli,j} - X_{i,j}(t))$$

Position updates follow equation (2). The proposed strategy is an alternative improvement to the second and third terms of the standard PSO update formula, enabling the algorithm to explore appropriately while possessing excellent exploitation capability. Due to CDL's learning samples providing strong guidance for particle search directions, the overall exploitation capability is superior, making it suitable for updating slave groups in the cluster structure. The master group is responsible for large-scale exploration in the search space, thus selecting CLS with strong global search capability. The specific update process is detailed in the algorithm flowchart [Figure 3: see original paper].

2.3 Multi-Swarm Collaboration

The master group explores the large-scale search space, obtaining high-quality solutions while maintaining good population diversity, but slowing convergence speed. Slave groups are responsible for high-precision exploitation near better solutions, greatly accelerating convergence, but if the nearby global best is a local optimum, particles cannot escape this region. Therefore, information from slave groups is needed to guide the master group to accelerate convergence, and information from the master group is needed to help slave group particles escape local optima. Thus, information exchange between master and slave groups is necessary. The exchange process must consider four issues: (1) migration period;

(2) exchange objects; (3) information to exchange; (4) integration strategy. To address these issues, a new Solution Exchange Mechanism (SEM) is proposed.

(1) Migration Period: This represents how many generations the master and slave groups independently execute their respective algorithms before beginning information exchange and cooperation. Any algorithm optimizing a specific problem must trade off between solution accuracy and computational efficiency. If information exchange is too frequent, algorithm runtime increases significantly, reducing efficiency. If exchange intervals are too long, solution quality suffers. In SEM, the migration period is set to 7 generations.

(2) Exchange Objects: Exchanging among too many clusters affects algorithm efficiency, while exchanging among too few increases the risk of falling into local optima and affects solution quality. Since slave groups are mainly responsible for high-precision exploitation near local optima and all execute DLS, if they fall into local optima, information from the exploration-capable master group can help them escape. Therefore, exchange among slave groups is unnecessary as it would increase runtime. SEM only exchanges information between the master group and slave groups.

(3) Information to Exchange: Information typically exchanged among clusters can include global best, local best, and particle neighbor best. In CDL-MCPSO, the master and slave groups only involve learning from other particles during their respective iterations. The cooperation process is only used to improve the "local optimum" situation that may occur in slave groups and the slow convergence of the master group due to strong diversity. Therefore, neighbor best information need not be exchanged—only the master group's global best and each slave group's local best information are exchanged.

(4) Integration Strategy: This processes the exchanged information to guide subsequent particle searches. The detailed process of the entire solution exchange mechanism is summarized as follows:

Master Group Update Strategy: After the master and slave groups independently execute their respective algorithms for 7 generations, all slave groups send their local bests $S_i^{lbest}(i = 1, 2, 3, 4)$ to the master group. The master group updates particles based on: (1) selecting the best among all slave group local bests and its own global best; (2) using this selected best to update particles according to equation (1). This update strategy adjusts particle trajectories based on competition between master and slave groups. The velocity update formula is:

$$V_{i,j}(t+1) = \omega V_{i,j}(t)+c_1 r_{i,j}(X_{MM,j}^{gbest}-X_{i,j}(t))+c_2 r_{2,j}(X_{SM,j}^{gbest}-X_{i,j}(t))+c_3 r_{3,j}(P_{i,j}^{best}-X_{i,j}(t))$$

where $P_{i,j}^{best}$ is the collaborative factor, and position updates still use equation (2).

Slave Group Update Strategy: Compare all slave groups' global best S^{gbest} with the master group's global best M^{gbest}, select the better one to replace a randomly chosen particle in all slave groups. This allows slave groups to indirectly share information without significantly affecting algorithm efficiency. The proposed new solution exchange mechanism not only balances algorithm exploration and exploitation but also accelerates convergence speed while greatly improving solution quality. and respectively show the solution exchange mechanism and multi-group collaboration related variables.

2.4 Latin Hypercube Sampling-Based Initialization Strategy

Latin Hypercube Sampling was first proposed by McKay [24] in 1979. Compared with ordinary sampling methods like Monte Carlo, it divides the input search space into equal intervals and randomly extracts samples from each interval, forcibly representing values from each interval, thus featuring uniform distribution. It ensures better efficiency than ordinary samples with fewer samples. [Figure 2: see original paper] shows a schematic diagram of Latin Hypercube Sampling in 2D search space. Samples generated through Latin Hypercube Sampling are better uniformly distributed throughout the space, effectively avoiding premature convergence. The specific sampling steps are:

Step 1: Determine sampling scale H;
Step 2: Divide each dimension variable X_i's domain interval $[X_{il}, X_{ih}]$ into H equal subintervals, producing H^n small hypercubes;
Step 3: Generate an $H \times n$ matrix A, where each column of A is a random permutation of the sequence $\{1, 2, ..., n\}$;
Step 4: Each row of A selects only one small hypercube, generating one sample within each selected small hypercube, thus selecting H distinct samples.

[Figure 2: see original paper] and [Figure 3: see original paper] respectively show the Latin Hypercube Sampling schematic in 2D space and the proposed algorithm flowchart.

---

3 Rationality Analysis of CDL-MCPSO Algorithm

3.1 Rationality of Removing Global Best Term Learning in CLS

Since CDL is developed based on DLS's velocity update formula, we first need to analyze the problems in DLS's velocity update strategy and then explain why CDL improvement is necessary. The analysis is as follows: Like most PSO variant velocity update formulas, DLS's velocity update formula also contains three parts: inertia part $\omega V_{i,j}(t)$, personal best learning part $c_1 r_{i,j}(X_{dli,j} - X_{i,j}(t))$ that constructs learning samples by learning from global best, and sample learning part $c_2 r_{2,j}(gbest_j - X_{i,j}(t))$. We generally need to assign exploration and exploitation capabilities to the last two terms of the formula. The last term's learning of global best $gbest_j$ undoubtedly assigns exploitation capability. Under the premise that the population already has exploitation capability, if the second term's personal best learning object still selects global best $gbest_j$, which further enhances exploitation capability, even though the new position obtained after personal best learns from global best is not global best, the learning process is dimension-by-dimension. Each learning does not necessarily involve only one dimension. When the new position's fitness value is better than personal best's fitness value, the learning process stops. Even in high-dimensional problems, after several generations of learning, the sample obtained by the second term of the formula will eventually become global best rather than a value between personal best and global best. At this point, for subsequent particle updates, the last two terms of the formula will become completely identical, doubling the exploitation capability assigned to the last term due to the proximity of the last two terms. If the global best found by particles is a local optimum and there is no learning from other particles during iteration, the population will undoubtedly face the risk of falling into local optimum without escape conditions. It should be noted that the iteration count experienced when the second term's learning sample becomes global best only accounts for a small proportion of the maximum iteration count set in this paper.

To address the above analysis deficiencies, this paper proposes the CDL improvement strategy for the following reasons: First, to weaken the double exploitation capability that particles possess during most subsequent iterations, one term (learning from global best) needs to be removed while introducing strategies to enhance exploration capability. The CLS strategy developed for multimodal problems has proven effective in exploration, capable of finding optimal solutions in optimization problems with multiple peaks. The reason for removing the global best learning term rather than the constructed learning sample term is that during initial iterations, to avoid missing any promising regions that may contain optimal solutions, we want the algorithm's overall exploration capability to be greater than exploitation capability. The retained learning sample term is not global best in the initial stage, so its traction on particles is not obvious, allowing particles in the population to explore the search space maximally. In later stages, after the learning sample becomes global best, it can completely replace the removed global best term to 发挥其开发能力, enabling the algorithm to maximize exploration or exploitation capabilities in different iteration periods. In summary, removing the global best term learning in the CDL strategy is reasonable.

3.2 CDL-MCPSO's Ability to Overcome Over-Exploitation and Theoretical Basis

CDL-MCPSO's theoretical basis for overcoming over-exploitation mainly includes three aspects: (1) Multi-swarm strategy based on master-slave paradigm—can compensate for the disadvantage of single populations having no other population information to guide when over-exploitation occurs; (2)

Slave group execution strategy—adds learning from other particles while canceling half the exploitation capability, making over-exploitation less likely; (3) Support from Solution Exchange Mechanism (SEM)—even if over-exploitation leads to falling into local optimum, information exchange and cooperation can help achieve escape. Specific explanations are as follows:

Compared with single populations, master-slave paradigm-based multi-subgroups have better diversity, reducing the possibility of particles falling into local optima due to over-exploitation because information from other populations can be used for guidance. Second, in the selection of algorithms for each subgroup, the master group executes the comprehensive learning strategy (CLS), which has significant optimization effects on multimodal problems, endowing the master group with large-scale exploration capability. Obviously, the master group is not prone to over-exploitation. For each slave group, this paper executes the optimized DLS—CDL—focusing on endowing the algorithm with exploitation capability while adding the part that learns from all other particles, making over-exploitation even less likely. Even if it occurs, the newly proposed SEM can increase population diversity through information exchange, which can also be described as avoidance or improvement of possible over-exploitation situations.

3.3 Changes in CDL-MCPSO's Optimization Capability at Different Iteration Stages and Balance Between Exploration and Exploitation

In the early iteration stage, the population is divided into multiple subgroups and initialized using Latin Hypercube Sampling, ensuring uniform distribution of population particles in the search space with good diversity and excellent exploration capability. During iteration, the master group executes CLS whose learning samples are particles' personal bests learning probabilistically from all other particles' personal bests, synthesizing excellent information found by all other particles so far. Although the slave group's velocity update formula includes the personal best learning term toward global best, in the early iteration stage, the guidance of slave group learning samples $X_{dli,j}$ on particles is less obvious compared to $X_{cl,i,j}$ and the master group's exploration-focused learning strategy, ensuring the population can conduct sufficient exploration in the search space.

In the middle iteration stage, each particle's personal best value has significantly improved compared to the early stage, resulting from the excellent exploration capability in the early stage. At this point, for both master and slave groups, since the learned objects' solutions have improved, the learning sample values also improve accordingly. The entire population's updates slowly approach the final optimal solution, with both exploration and exploitation capabilities coexisting, optimizing solution quality without causing premature convergence while retaining some exploration capability to continue exploring potentially better solutions.

In the later iteration stage, as the solution quality found by the population further improves, the slave group learning sample $X_{dli,j}$ values become increasingly close to and eventually equal the global best. Although the other learning sample $X_{cl,i,j}$ is obtained by learning from other particles' personal bests, since the iteration process has reached the later stage of the entire iteration cycle, each particle's personal best solution quality is already very close to the problem's final solution. At this point, the value fluctuation of $X_{cl,i,j}$ gradually decreases, and the population no longer conducts large-scale exploration but mainly optimizes around the currently found best solution. Because of the good exploration capability in the early stage and appropriate exploration capability in the middle stage, particles have already conducted high-level exploration of the problem space. Therefore, in the later stage when particles approach the global best, there is no situation where they approach a local optimum. Thus, the entire population mainly 发挥开发作用 in the later stage.

In summary, the improved PSO algorithm's optimization capability can 发挥勘探或开发能力 or both simultaneously according to different needs at different iteration stages, achieving an effective balance between the two.

3.4 Convergence Analysis of CDL-MCPSO Algorithm

Assumption 1 If $f(D(x,\zeta)) \le f(\zeta)$ and if $\zeta \in S$, then $P(x \in S) = 1$.

Assumption 2 For any subset A of S, if its measure $v(A) > 0$, let M be the n-dimensional closure of subset A, and A' be the set generated by measure $v(A)$.

Theorem 1 Assuming the objective function f is a measurable function, the problem's constraint space S is a measurable subset, and $\{x_t\}$ is the solution sequence generated by a random algorithm, then when Assumptions 1 and 2 are satisfied:

$$\lim_{t\to\infty} P[x_t \in R_\varepsilon] = 1$$

where R_ε is the set of global optimal points. Therefore, for any random optimization algorithm, as long as it satisfies Assumptions 1 and 2, it can be guaranteed to converge to the global optimal solution with probability 1.

Theorem 2 The CDL-MCPSO algorithm can converge to the global optimal solution with probability 1.

Proof: CDL-MCPSO satisfies Assumption 1. Let the iteration function D of CDL-MCPSO be represented as $D(x,\zeta) = \zeta$ if $f(x) \le f(\zeta)$, otherwise $D(x,\zeta) = \zeta$. It is easy to prove that CDL-MCPSO satisfies Assumption 1.

Proof: CDL-MCPSO satisfies Assumption 2. Let the union of sample spaces of a particle swarm of size N be $\bigcup_{i=1}^{N} M_{i,t}$, which must contain S, i.e., $S \subseteq \bigcup_{i=1}^{N} M_{i,t}$. For standard PSO, let $M_{i,t} = \{x_{i,t}\}$, and other particles satisfy

$M_{j,t} = \{x_{j,t}\}$. Substituting the velocity update formula into the position update formula yields:

$$x_{i,t+1} = x_{i,t} + \omega(x_{i,t} - x_{i,t-1}) + \varphi_1(p_{i,t} - x_{i,t}) + \varphi_2(p_{g,t} - x_{i,t})$$

where $0 \leq \varphi_1 \leq c_1$ and $0 \leq \varphi_2 \leq c_2$. When $\varphi_1 = \varphi_2 = 0$, we have $x_{i,t+1} = x_{i,t} + \omega(x_{i,t} - x_{i,t-1})$. Clearly, $M_{i,t}$ represents a hyper-rectangle with one endpoint at $x_{i,t}$ and the other at $x_{i,t-1}$. As iteration number t increases, $diam(M_{i,t}) \to 0$. Thus, $\bigcup_{i=1}^{N} M_{i,t}$ continuously decreases, and there exists an integer t' such that when $t > t'$, there exists a set A with $\mu(A) = 0$. This indicates that standard PSO does not satisfy Assumption 2 and cannot converge to the global optimal solution with probability 1.

CDL-MCPSO generates new particles using multi-subgroup, different learning strategies, and inter-population collaborative exchange mechanisms based on standard PSO. Therefore, for normally evolving particles, let the union of their support sets be $\bigcup_{i=1}^{N} M_{i,t}^{v}$; for particles generated using the above improvement strategies, let the union of their support sets be $\bigcup_{i=1}^{N} M_{i,t}^{s}$. Due to the guidance of improvement strategies, there must exist an integer t_1 such that when $t > t_1$, $\bigcup_{i=1}^{N} M_{i,t}^{s} \supseteq S$. Therefore, for CDL-MCPSO, there exists an integer t_2 such that $\bigcup_{i=1}^{N} M_{i,t} \supseteq S$ for any $t > t_2$. Thus, CDL-MCPSO satisfies Assumption 2.

Conclusion 1 CDL-MCPSO is a globally convergent algorithm.

Proof: Let $\{p_{g,t}\}$ be the solution sequence generated by CDL-MCPSO. Since CDL-MCPSO satisfies Assumptions 1 and 2, through Theorem 1 we know:

$$\lim_{t\to\infty} P[p_{g,t} \in R_\varepsilon] = 1$$

holds, thus CDL-MCPSO can converge to the global optimal solution with probability 1. The algorithm's convergence speed and solution accuracy are analyzed in detail in the experimental section.

3.5 Time Complexity Analysis

Traditional PSO mainly includes three parts: initialization, fitness evaluation, and velocity/position updates, with complexities of $O(2\times N\times D)$, $O(N\times D)$, and $O(2 \times N \times D)$ respectively (N and D represent population size and dimension). Therefore, PSO's time complexity is $O(N \times D)$. Compared with traditional PSO, CDL-MCPSO needs to construct learning samples and evaluate all slave group local bests during master-slave group information exchange. Since master-slave groups only exchange information when continuous iteration count reaches the refresh gap, and the evaluated local bests come from four slave groups with only four sets of data evaluated per exchange, the multi-swarm collaboration process has time complexity $O(3)$. Moreover, learning samples only need reconstruction when a particle's personal best updates or hasn't updated for several generations. The worst-case time complexity for constructing learning samples is $O(N \times D)$, and the complexity for judging whether iteration reaches refresh gap and whether to stop is $O(2)$. Therefore, CDL-MCPSO's worst-case time complexity is also $O(N \times D)$, including initialization $O(2 \times N \times D)$, evaluation $O(N \times D + 5)$, and updates $O(2 \times N \times D + N \times D)$. Based on the above analysis, CDL-MCPSO's time complexity is at the same level as the basic PSO algorithm.

---

4 Experimental Validation and Analysis

4.1 Individual Distribution and Population Diversity Analysis at Typical Iteration Stages

[Figure 4: see original paper]–[Figure 6: see original paper] show individual distribution maps of the population at the 50th, 500th, and 900th generations.

[Figure 4: see original paper] shows the approximate distribution in the early iteration stage. The master and slave groups independently execute CLS and DLS. CLS's learning objects are all other particles' personal bests. Although DLS's second term learns from the global best, the obtained sample requires several iterations to become the global best. Population diversity is good, with particles basically scattered uniformly in the search space.

[Figure 5: see original paper] shows the approximate distribution in the middle iteration stage. The slave groups iterate DLS formula's second term to obtain samples close to the global best, but differences in other particles' personal bests in the master group still maintain good diversity in learned samples. Part of DLS's exploitation capability is neutralized, and with population information exchange at certain iteration intervals, particles only slowly begin to converge while still maintaining good population diversity.

[Figure 6: see original paper] shows the approximate distribution in the later iteration stage. The slave groups' learning samples $X_{dli,j}$ play an exploitation role, population diversity weakens, and most particles gather near the global best solution. However, the existence of CLS still scatters individual particles not too far from the global best solution.

4.2 Test Functions

To verify CDL-MCPSO's performance, ten functions from the Benchmark test function set are used for validation, including two unimodal functions (f_1–f_2), three composition functions (f_3–f_5), and five multimodal functions (f_6–f_{10}). Test function details are shown in .

4.3 Comparison Algorithms

To verify CDL-MCPSO's performance in solving complex problems, it is compared with five improved PSO algorithms: Unified Particle Swarm Optimization (UPSO) [13], Fully Informed Particle Swarm (FIPS) [11], Particle Swarm Optimization based on Dimensional Learning Strategy (DLPSO) [21], Parallel Comprehensive Learning Particle Swarm Optimization (PCLPSO) [17], and Standard Particle Swarm Optimization (PSO-W) [2]. To ensure fair testing, all algorithms use identical parameter settings: population size of 40, iteration count of 1000, and each algorithm independently runs 30 times on each test function under three dimensions ($D = 10, 30, 50$). Experimental environment: AMD Ryzen 5 3500H with Radeon Vega Mobile Gfx 2.10 GHz, RAM 16GB, Windows 10 OS, Matlab R2016b. Parameter settings for CDL-MCPSO and comparison algorithms are shown in .

4.4 Algorithm Performance Analysis

– show the mean and standard deviation of each algorithm on test functions for $D = 30$ and $D = 50$. The best results among the six algorithms are highlighted in bold. The data shows that CDL-MCPSO achieves better or equivalent optimization results on most functions, whether unimodal, multimodal, or composition functions, except for slightly insufficient performance on test function f_4. Stability is slightly inadequate on individual functions. The main reason for these results is that the comprehensive learning strategy (CLS) and comprehensive dimensional learning strategy (CDL) employed by master-slave groups demonstrate excellent exploration capability in the early iteration stage, exploring all regions that may contain better solutions and laying the foundation for improving final solution quality.

[Figure 7: see original paper]–[Figure 12: see original paper] respectively show convergence curves of CDL-MCPSO on functions f_1 (unimodal), f_4 (multimodal), and f_9 (composition) under 30 and 50 dimensions. For function f_1's convergence curves, [Figure 7: see original paper] shows that the improved algorithm significantly outperforms PSO-W, UPSO, and FIPS. Compared with PCLPSO, the improved algorithm finds the global optimum, while PCLPSO not only fails to find the global optimum but also begins converging around generation 50, with overly fast convergence. [Figure 10: see original paper] shows similar situations to [Figure 7: see original paper]. For function f_4's convergence curves, especially [Figure 11: see original paper], CDL-MCPSO's performance is significantly superior to other comparison algorithms. From function f_9's convergence curves, CDL-MCPSO not only finds obviously better solutions than other comparison algorithms but also converges faster, without "oscillation" in later iterations, solving the "dimension degradation" problem. Although CDL-MCPSO also has shortcomings—algorithm stability is slightly inferior to other algorithms on individual test functions—overall, CDL-MCPSO shows substantial improvement in optimization results compared with other algorithms.

To further verify CDL-MCPSO's performance, statistical analysis methods are employed using Wilcoxon rank-sum test with significance level 0.05 to judge algorithm performance. The symbols "+", "−", and " " respectively indicate that CDL-MCPSO's results are better than, worse than, or equivalent to the corresponding algorithm's results. From 's Wilcoxon test results, at $\alpha = 0.05$, CDL-MCPSO shows certain advantages over comparison algorithms on test functions.

Based on the above comparative analysis, CDL-MCPSO can maintain better diversity to explore the original space for high-quality solutions and possesses the best capability to reach optima for rapid convergence. Therefore, CDL-MCPSO is a feasible method to effectively improve PSO performance.

5 Conclusion

To improve algorithm efficiency and balance exploration and exploitation, this paper proposes the Comprehensive Dimensional Learning-based Multi-Swarm Collaborative Particle Swarm Optimization algorithm (CDL-MCPSO) based on master-slave paradigm cluster structure. Master and slave groups independently execute learning strategies with different functions, where the slave group learning strategy is developed to address the "over-exploitation" problem in DLS. To improve solution quality, a new solution exchange mechanism is also proposed for information sharing among groups. Experimental results prove that CDL-MCPSO not only finds better or equivalent optimization results on most unimodal, multimodal, and composition functions but also shows obvious advantages in convergence speed and solution quality, solving the "oscillation" and "dimension degradation" problems while effectively avoiding "over-exploitation." Future work will mainly focus on improving algorithm stability on some test functions and applying it to practical complex problems for effectiveness validation.

References

[1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory [C]// Proc of the 6th International Symposium on Micro Machine and Human Science. Japan: IEEE, 1995: 39-43.

[2] Shi Y, Eberhart R. A modified particle swarm optimizer [C]// IEEE International Conference on Evolutionary Computation Proceedings. Piscataway NJ: IEEE, 1998: 69-73.

[3] El Afia A, Sarhani M, Aoun O. Hidden markov model control of inertia weight adaptation for particle swarm optimization [J]. IFAC-PapersOnLine, 2017, 50 (1): 9997-10002.

[4] Gou Jin, Lei Yuxiang, Guo Wangping, et al. A novel improved particle swarm optimization algorithm based on individual difference evolution [J]. Applied Soft Computing, 2017, 57: 468-481.

[5] Nasir M, Das S, Maity D, et al. A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization [J]. Information Sciences, 2012, 209: 16-36.

[6] Jiao Chongyang, Zhou Qinglei, Zhang Wenning. Particle swarm optimization with mixed topology and its application in test data generation [J]. Computer Science, 2017, 44 (12): 255-260.

[7] Gong Yuejiao, Li Jingjing, Zhou Yicong, et al. Genetic learning particle swarm optimization [J]. IEEE Trans on Cybernetics, 2015, 46 (10): 2277-2290.

[8] Aydilek I B. A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems [J]. Applied Soft Computing, 2018, 66: 232-249.

[9] Şenel F A, Gökçe F, Yüksel A S, et al. A novel hybrid PSO-GWO algorithm for optimization problems [J]. Engineering with Computers, 2019, 35 (4): 1359-1373.

[10] Chen Yuegang, Xu Yi. Multi-objective decomposition particle swarm algorithm based on comprehensive learning strategy [J]. Microelectronics and Computer, 2018, 35 (10): 75-79.

[11] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler maybe better [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3): 204-210.

[12] Tian Mengdan, Liang Xiaolei, Fu Xiuwen, et al. Multi-subgroup particle swarm optimization algorithm with game probability selection [J]. Computer Science, 2021, 48 (10): 67-76.

[13] Parsopoulos K E, Vrahatis M N. UPSO: A unified particle swarm optimization scheme [C]// International Conference of Computational Methods in Sciences and Engineering. Greece Florida: CRC Press, 2019: 1-7.

[14] Yuan Xiaoping, Jiang Shuo. Improved particle swarm optimization algorithm based on hierarchical autonomous learning [J]. Computer Applications, 2019, 39 (01): 148-153.

[15] Li Dongyang, Guo Weian, Lerch A, et al. An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization [J]. Swarm and Evolutionary Computation, 2021, 60: 100797.

[16] Zhou Shangbo, Sha Long, Zhu Shufang, et al. Adaptive hierarchical update particle swarm optimization algorithm with a multi-choice comprehensive learning strategy [J]. Applied Intelligence, 2022, 52 (2): 1217-1235.

[17] Gulcu Ş, Kodaz H. A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization [J]. Engineering Applications of Artificial Intelligence, 2015, 45: 33-45.

[18] Liao Weilin, Cheng Shan, Shang Dongdong, et al. Multi-strategy fusion particle swarm optimization algorithm [J]. Computer Engineering and Applications, 2021, 57 (01): 69-76.

[19] Ji Wei, Li Yingmei, Ji Weidong, et al. PSO algorithm of dual-population comprehensive learning based on particle replacement [J]. Journal of Computer Science and Exploration, 2021, 15 (04): 766-776.

[20] Xu Guiping, Cui Quanlong, Shi Xiaohu, et al. Particle swarm optimization based on dimensional learning strategy [J]. Swarm and Evolutionary Computation, 2019, 45: 33-51.

[21] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Trans on Evolutionary Computation, 2006, 10 (3): 281-295.

[22] Parsopoulos K E, Vrahatis M N. On the computation of all global minimizers through particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3): 211-224.

[23] Van Den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization [J]. IEEE Trans on Evolutionary Computation, 2004, 8 (3): 225-239.

[24] Mckay M D, Beckman R J, Conver W J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code [J]. Technometrics, 2000, 42 (1): 55-61.

[25] Solis F J, Wets R J B. Minimization by random search techniques [J]. Mathematics of Operations Research, 1981, 6 (1): 19-30.

Note: Figure translations are in progress. See original paper for figures.