International Journal of Engineering Research And Management (IJERM)
ISSN: 2349- 2058, Volume-11, Issue-05, May 2024

Evolutionary Multi-task Optimization Algorithm Based On Prior Knowledge

Gaomin Yin

***Abstract*—Traditional evolutionary algorithms are often used to solve single-task optimization problems. With the study of evolutionary algorithms, it is found that most optimization tasks often have potential correlation, which indicates that the knowledge obtained in the evolution process of one optimization task can be used to optimize another task to further optimize the performance of the target task. With the rapid development of machine learning technology, the idea of making use of commonalities or differences among multiple tasks for efficient learning has been widely studied in the field of multi-task learning.In order to strengthen the positive transfer of knowledge, we can consider using strong correlation prior knowledge to construct help tasks and optimize the original tasks together. In this paper, a method of constructing help tasks is proposed, which is based on the original problem and constructed by multi-objective decomposition of sub-problem groups. These sub-problem sets are closely related to the original problem and belong to the strongly correlated tasks. Therefore, the help tasks constructed by this method can promote the forward knowledge transfer of multi-task optimization theoretically. The experimental results show that the efficiency of the proposed algorithm is improved significantly in task optimization.**

***Index Terms*—knowledge transfer, helper task , piror konwledge, multitask optimization algorithm**

I. Introduction

Evolutionary algorithm (EA), which simulates the process of biological evolution, is a heuristic search algorithm proposed under the influence of Darwinian evolution, and uses selection, crossover and mutation to realize the evolutionary process of the population. The evolutionary algorithm generates new child individuals through continuous iteration, and then selects more excellent individuals to enter the next iteration process, and obtains the optimal solution through continuous iteration search. With strong parallelism, simple implementation method and strong search ability, evolutionary algorithm has been widely used to solve real life optimization problems, including path planning[1][2], network security, intelligent scheduling [3]and other problems.

With the rapid development of science and technology, modern society is faced with more and more complex optimization problems, such as energy scheduling, traffic planning, production scheduling and so on. These problems are usually highly nonlinear, multi-modal, multi-constrained and dynamic. Traditional evolutionary algorithms can only solve one task at a time, which makes it difficult for traditional optimization algorithms to solve effectively. However, in these problems, there are often similarities between many tasks. Knowledge sharing is carried out between them, and the knowledge used to solve one task can also be used to solve another task, which can improve the optimization efficiency between tasks. Inspired by multi-task learning, Gupta et al. [4] combined multi-task learning with evolutionary algorithm for the first time and proposed Multifactorial Evolutionary Algorithm (MFEA). Evolutionary multitasking algorithm, as a new evolutionary paradigm, can deal with multi-task optimization problems across domains and optimize each task in parallel. Fig.1 schematic diagram of multi-task optimization algorithm processing. In real life, many problems can be represented as multi-task optimization problems, such as path planning problems[5], cloud service composition problems[6], complex optimization problems [7],which can be solved by evolving multi-task optimization algorithms.

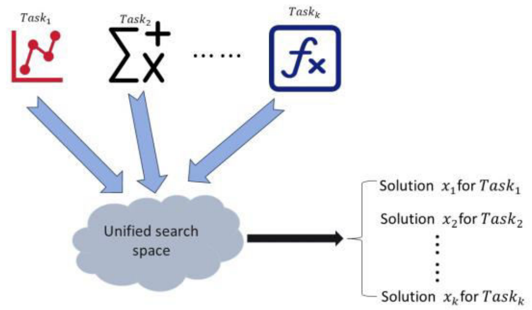


Fig.1 schematic diagram of multi-task optimization algorithm processing.

In multitasking Optimization (Multi - Task Optimization, MTO) algorithm is put forward before, scholars complex Optimization problem was divided into two categories: The first type is Single-Objective Optimization (SOO) and the second type is Multi-Objective Optimization (MOO). Single-objective optimization means that when the function to be optimized has only one objective value, the optimal solution satisfies the constraints of the function to obtain the variable corresponding to the maximum or minimum value of the function, while multi-objective optimization requires consideration of multiple objective functions at the same time. In most cases, there are conflicts between these objectives, and the optimization of one objective is often accompanied by the disadvantage of the other objective. So in this case you need to consider these compromises, which are called Pareto optimal solutions, not optimal solutions for

Manuscript received May 15, 2024
Gaomin Yin, School of Computer Science and Technology, Tiangong University, Tianjin, China

a particular goal. Until the MTO was proposed, it became a new paradigm. Although both multi-task optimization and multi-objective optimization seek solutions to a set of objective functions, the difference is that multi-objective optimization is only a single task. Multi-task optimization has multiple tasks, and there is some potential parallelism between these tasks in the optimization process, which can obtain some useful knowledge, realize knowledge transfer of cross-domain tasks, and accelerate the convergence of tasks. To find the optimal solution to the task. Multi-objective multi-task optimization algorithm combines multi-objective optimization problems with multi-task optimization algorithms, and introduces NSGA-II on the basis of MFEA. Each optimization task has an independent search space. The proposed algorithm makes further development in solving optimization problems and can provide better task solutions.

The research of evolutionary multi-task optimization algorithm will promote the technological innovation in related fields. With the development of big data, cloud computing[8], artificial intelligence and other technologies, multi-task optimization algorithms have great potential in dealing with complex problems and large-scale data.Transform the search space, such as LDA-MFEA[9].

Due to the remarkable effect achieved by MFEA, many scholars have proposed a variety of variants based on it. Zheng et al.[10] proposed a method that adaptively adjusts the degree of knowledge transfer across task domains, taking into account the different degree of correlation between different tasks, so that knowledge capture between tasks can be realized to a large extent and useful knowledge can be shared to a large extent. Han et al.[11] introduced decision space and target space based on each task, used knowledge estimation metric to evaluate the knowledge carried by particles to improve the effectiveness of solving each task, and introduced a parameter that could be automatically adjusted to realize self-regulated transfer mode and adjust the intensity of knowledge transfer. Shi et al. [12] designed an improved assortative mating operation based on DE/rand/2 mutation in order to make full use of the optimal solution generated by each generation in assortative mating. The improved operation generates offspring for each individual by running perturbations around the current optimal individual. Integrated opposition-based search strategies to balance the development and exploration of each search space. The idea of Wu et al.[13] is to estimate the bias between the two tasks and then eliminate it in chromosome transfer so that the optimal solutions of the two tasks are close to each other. In this way, a promising chromosome in one task can also be transformed into a promising solution for another task, thereby speeding up the rate of convergence and efficiently transferring the most suitable chromosome.

II. PRELIMINARIES

A. Multifactorial optimization problem

Multifactorial Optimization (MFO) is an evolved multi-task paradigm. It is characterized by the simultaneous existence of multiple search spaces corresponding to different tasks. Each search space is unique. During the optimization process, each Each task contributes different information that affects the task optimization process. The problem of MFO can be defined as follows:

$$\{x_1, x_2, ..., x_k\} = \arg\min \{f_1(x), f_2(x), ..., f_k(x)\} \tag{1}$$

Where x_i is the feasible solution of the task $f_i(x), i=1,2,...,k$. Each individual x_i in MFEA has the following four characteristics:

(1)Factorial Cost: for a given individual x_i and task T_j, the factor cost of the individualis calculated by evaluating the objective function value and the total violation value on the corresponding task , defined as Ψ_j^i.

(2)Factorial Rank: The factorial rank r_j^i of the individual x_i is the index value obtained by sorting the factor cost Ψ_j^i in ascending order when evaluating the task T_j.

(3)Scalar Fitness: The scalar fitness φ_i of an individual x_i is the reciprocal of its factor ranking on the task for which it performs best in all task evaluations $\varphi_i = 1/\min_{j\in\{1,2,...,k\}}\{r_j^i\}$.

(4)Skill Factor: The skill factor of individual x_i, τ_i is the index of x_i of the task in which the smallest factor rank can be obtained, $\tau_i = \arg\min_{j\in\{1,2,...,k\}}\{r_j^i\}$.

B. Unified encoding and decoding methods

In MFO, each task to be optimized has different spatial characteristics, so knowledge transfer between tasks cannot be realized directly.In order to realize knowledge transfer between task domains, MFEA performs uniform encoding and decoding operations for optimization tasks.All the individuals in the population by encoding operation to build a unified search space, with a different search space mission after the searching space of the individual coding to unity, knowledge transfer between tasks to improve the optimization efficiency of the algorithm. The decoding operation is to map the individual in the unified search space back to the original solution space to obtain the optimal solution of the task.

Suppose that K tasks are optimized at the same time and the search space corresponding to the j_{th} task is D_j, then the dimension of the unified search space is the largest dimension among all tasks, that is $D_{Multitask} = \max_{j\in\{1,2,...,k\}}\{D_j\}$, which can better realize the information sharing between tasks. As shown in Fig.2, individuals in the unified search space encode the solution of three chromosomes. The spatial dimensions corresponding to the three target tasks are respectively 6,7, and 9. Each task performs coding operations, and the dimension of the unified search space obtained is 9. The decoding operation is to decode the individual of each task into the corresponding original task for evaluation. Suppose that an individual p_i in the population corresponds to the task T_j. In the evaluation, $x_j = L_j + p_i \times (U_j - L_j), j \in \{1,2,...,k\}$,maps the individual p_i decoding to the search space corresponding to the original task T_j. Where x_j represents the candidate solution corresponding to task T_j, U_j and L_j table the upper and lower bounds of the search space of task T_j respectively.

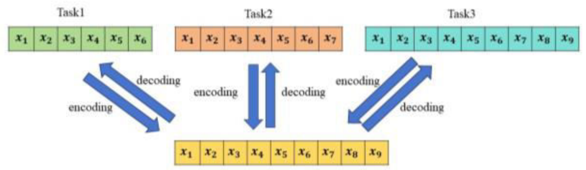


Fig. 2 Individual representation in the unified search space

C. MFEA algorithm framework

There are two key processes in MFEA: assortative mating and vertical cultural transmission. Assortative mating is the process of producing offspring, and vertical cultural transmission assigns skill factors to individuals and evaluates them. In the generation of offspring, the process of information transfer needs to be controlled by random mating probabilities(*rmp*).

Algorithm 1:MFEA algorithm framework

Input: population size *N*, maximum number of iterations *Gen*, *K* optimization tasks
Output:the optimal solution for each task
1.Initialize population *P*, contains *N* individuals
2.Each individual in the population is randomly assigned a skill factor τ_i,and evaluate the individuals
3.t=0
4.**while(t<Gen)**
5. Generation of offspring population *C* through assortative mating
6. Assign skill factors for individuals in offspring population *C* through vertical cultural communication and evaluate
7. Merge parent and offspring populations to update scalar fitness and skill factors for all individuals
8. Select *N* elite individuals from the merged population to form a new population and enter the next generation
9. t=t+1
10.**end**

In the process of assortative mating, knowledge transfer is carried out in a unified search space, and individuals between different tasks perform crossover operation, so as to realize knowledge sharing between tasks and achieve the purpose of knowledge transfer. If individuals have the same or similar skill factors, it is easier to perform cross-operation. In this case, individuals p_a and p_b cross freely to generate offspring, so that both offspring individuals have the genetic information from the two parents of different tasks, realizing knowledge transfer.

When choosing whether to perform crossover operation, it is also determined by random mating probability, the size of rmp is between 0 and 1. If rmp value is closer to 0, it means that only individuals with the same cultural background can freely perform crossover operation. If the value of rmp is closer to 1, it means that individuals can randomly mate to produce offspring.

In the process of vertical cultural transmission, the genetic characteristics of individuals are influenced by their parents. If the offspring individual has two parents, randomly select one parent's skill factor to inherit. If there is only one parent, the child individual directly inherits the skill factor of its sole parent. At the same time, in order to reduce calculation, only the tasks pointed to by the skill factor are evaluated.

III. PROPOSED APPROACH

A. Research motivation

It has been proved that the correlation between multiple tasks will greatly affect the performance of MFEA. When the tasks are highly correlated, there is a lot of forward transfer of shared knowledge between tasks, which speeds up the convergence rate of the algorithm. Therefore, in order to strengthen the forward transfer of knowledge, we can consider using the strongly correlated prior knowledge helper task to optimize together with the original task. The helper task used in this paper is built on the sub-problem set of the multi-objective decomposition of the original problem. These sub-problem sets are closely related to the original problem and belong to the strongly correlated tasks. Therefore, the helper tasks constructed by this method can promote the positive knowledge transfer theory of MFEA. In this paper, a prior-knowledge-based helper-tasks (PKHT) algorithm for multi-task evolution based on Prior knowledge of helper tasks is proposed. In order to verify the effectiveness of MFEA-PKHT, which transforms prior knowledge into helper tasks, in improving the efficiency of MFEA in optimizing tasks, this paper designed experiments for data comparison and analysis, and the results proved that MFEA-PKHT, which uses prior knowledge helper tasks, performs extremely well in most multi-task optimization problems.

In this paper, we use a multi-objective decomposition method based on prior knowledge to construct strongly correlated helper tasks, and integrate this method into MFEA algorithm to enhance the forward knowledge transfer of the algorithm. The multi-objective decomposition of tasks is to optimize a single objective task by decomposing it into multiple objective tasks [14]. Multi-objective decomposition can decompose a problem into many small optimization problems that cover all the solution space of the original problem, so optimizing these subproblems simultaneously optimizes the original problem.

Based on MFEA, the MFEA-PKHT algorithm used in this study decomposes the original target task into multiple subtask groups according to decision variables or problem structure, and then constructs strongly correlated helper tasks. These decomposed sub-problems are closely related to the original problem, so they can promote the forward transfer of knowledge well and have a good optimization effect for MFEA.

B. Multiobjective decomposition of the original problems

Before starting the main algorithm work of MFEA-PKHT, the original task needs to be decomposed into multiple objectives. Common types of multipurpose decomposition construction assistance tasks include $f_1(x)+...+f_m$ conversion to $[f_1(x),...,f_m(x)]^T$, $f_1(x)+\lambda * f_2(x)$ is converted into $[f_1(x), f_2(x)]^T$.

In the existing multi-objective decomposition methods, there is less research on constructing helper tasks for continuous optimization tasks, and more research is related to combinatorial optimization tasks. In recent years, Lochtefeld et al. [15] proved the relationship between multi-objective decomposition and the formalization of helper tasks; Handl et al. [16]proved that multi-objective decomposition reduces the number of local optimizations and increases the size of incomparable solutions. Da et al. [17] proposed to add multi-objective decomposition as an helper task under the restrictive conditions of combinatorial optimization. Louis et al. [18]was the first to apply the helper task of multi-objective decomposition to the continuous optimization task.

These studies verify the positive effect of multi-objective decomposition on multi-task optimization, which can increase population diversity to avoid falling into local optimal, and provide better knowledge transfer. Each helper task in the MFEA-PKHT used in this paper is a strongly correlated helper task generated by the multi-objective decomposition method on the original task. Theoretically, the knowledge transfer of this strong correlation relationship is beneficial to the global convergence of the optimization algorithm. The experimental data also prove that using multi-objective decomposition to add prior knowledge to construct helper tasks can indeed enhance the algorithm. In this paper, we use the first method of decomposing the helper task, here decomposed into two sub-problems, $f_1(x)+...+f_2(x)$ converted to $[f_1(x),...,f_2(x)]^T$.

C. *The main flow of the algorithm*

Similar to MFEA, the transfer of genetic material from parent to offspring in MFEA-PKHT is also based on vertical cultural transmission, with crossover occurring easily among individuals of similar cultural backgrounds. The process of assortative mating is to produce offspring. Two parents are randomly selected from the parent population. If the two individuals have the same skill factor, they will perform crossover operation. If they do not have the same skill factor, then a random number from 0 to 1 determines whether they cross offspring or mutate their own offspring. The children obtained under the cross operation of the algorithm inherit the skill factor of one parent randomly, and the children obtained by variation inherit the skill factor of their unique parent.

Then the algorithm evaluates the fitness of individual offspring, and the algorithm execution the process of vertical culture transmission.Offspring and parents need to be merged first and then sorted, and Non-dominated sort and Crowding distance are applied to determine the ordering of new population individuals. According to the principle of quick sorting, the population is sorted by non-dominant rank. Firstly, the dominant number of each individual in the population and the dominant solution set of that individual are calculated, and the individuals with the dominant number of 0 in the population are placed in the non-dominant solution set of level 1. These classified individuals are then removed from the sorted set, while the dominance of other individuals is updated. The steps are repeated from the assigned individuals to the non-dominated solution set after the current non-dominated solution set is upgraded until all individuals are graded. Finally, the populations were sorted in ascending order according to the non-dominant ranking. Update skill factor τ_i and scalar fitness φ_i for all individuals. Finally, MFEA-PKHT uses the elite selection strategy to screen out individuals and select a new population P for the next iteration.

IV. SOO Experiment and Result Analysis

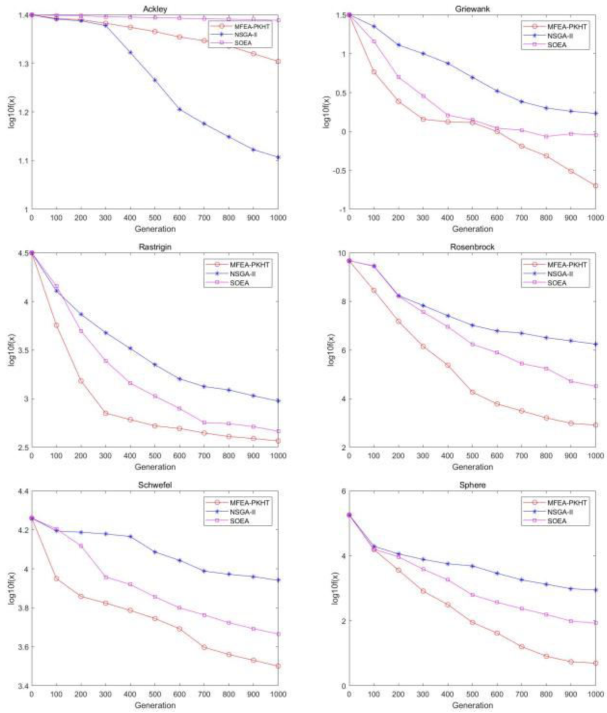
A. *Test set and parameter Settings*

In the SOO experiment, a SOO test set was tested to verify the effectiveness of the algorithm, consisting of seven single-objective optimization problems, namely Ackley, Griewank, Rastrigin, Rosenbrock, Schwefel, Sphere, and Weierstrass. The comparison algorithm used is SOEA and NSGA-II.

The parameters used in this experiment are set as follows:
(1) The experiment was processed by Wilcoxon rank sum test with a significance level of 5%.
(2) Random mating probability rmp=0.3 in MFEA-PKHT, SOEA and NSGA-II do not need to set random mating probability.
(3) SOEA solves the original single-objective problem f(x), NSGA-II solves F(x), which decomposes the original problem into a multi-objective optimization problem, and MFEA-PKHT solves the multi-task problem MTO.
(4) Population size is set to N=50.
(5) The maximum number of iterations is 1000.
(6) All the algorithms run independently 20 times.

B. *Experimental resultset*

Test set 1 is a single objective problem set consisting of 7 single objective problems. Fig.3 shows the convergence curves of the three algorithms. Through comparison, it can be seen that MFEA-PKHT can achieve faster convergence and better stability, indicating that the proposed multi-objective decomposition based on prior knowledge is effective for improving the diversity of the population and promoting the solution of the original problem. MFEA-PKHT only got slightly worse results on the two test problems Ackley and Weierstrass than SOEA and NSGA-II.



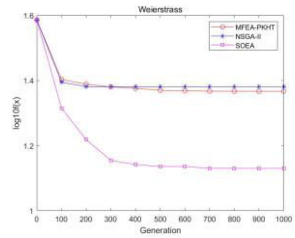


Fig.3 Convergence curve for obtaining the optimal average solution on 7 test problems

Compared with the other two comparison algorithms, MFEA-PKHT can solve the original target task and its helpertask task at the same time, and the knowledge transfer between the two tasks can solve the task faster, and the performance is also greatly improved. The added helpertask tasks are the result of the decomposition of the original problem by multipurpose, so the two tasks are closely related. In MFEA-PKHT, the multi-task processing method obtains some useful knowledge and helps to solve the original task through knowledge transfer while solving the helpertask task. In addition, it can help the task jump out of the local optimal, improve the convergence and diversity of the population.

By observing Fig.3, the proposed MFEA-PKHT is better than SOEA and NSGA-II in terms of overall algorithm performance when solving the single-objective problem set, among which Rastrigin and Schwefel problems can converge faster at the early stage of iteration. From Griewank's convergence curve, it can be seen that in the middle of the iteration, it falls into a local optimal situation, and then jumps out of the local optimal situation and continues to converge to a relatively excellent solution. This phenomenon shows that the introduced helpertask task is closely related to the original task, which is conducive to improving the efficiency of knowledge transfer between tasks and helping the task to escape the local optimal.

V. MTO EXPERIMENT AND RESULT ANALYSIS

A. *Test set and parameter Settings*

In order to verify whether MFEA-PKHT has advantages in multi-task optimization, this experiment uses an MTO test data set to compare the optimization tests of MFEA-PKHT, MFEA and MFEA-LDA. This paper adopts the MTO benchmark test question of CEC 2017 Evolutionary multi-task Optimization competition, which contains 9 questions. Each problem shown consists of a single objective optimization task, namely CIHS, CIMS, CILS, PIHS, PIMS, PILS, NIHS, NIMS, and NILS. According to the intersection degree of the optimal solution, the tasks can be divided into three categories: complete intersection (CI), partial intersection (PI), and no intersection (NI). Spearman rank correlation coefficient among landscapes was divided into high similarity (HS), medium similarity (PS), and low similarity (LS) according to the Spearman rank correlation coefficient Rs.

B. *Experimental resultset*

The convergence curves obtained by MFEA-PKHT, MFEA and MFEA-LDA algorithms on 9 benchmark problems are shown in Fig.4-Fig.5. Obviously, the proposed MFEA-PKHT shows better performance, showsthat the multi-objective decomposition strategy adopted is effective.

Fig.4 shows the comparison of the convergence curves of MFEA-PKHT, MFEA and MFEA-LDA in the three optimization task groups CIHS, CIMS, and CILS. It can be seen that MFEA-PKHT performs better than MFEA and MFEA-LDA in the optimization of these three groups of tasks, and the convergence speed is faster than the other two comparison algorithms.This indicates that the helpertask task constructed by the method proposed in this paper has a strong relevance, which generates a large amount of useful knowledge in the process of task optimization and promotes the positive transfer of knowledge.

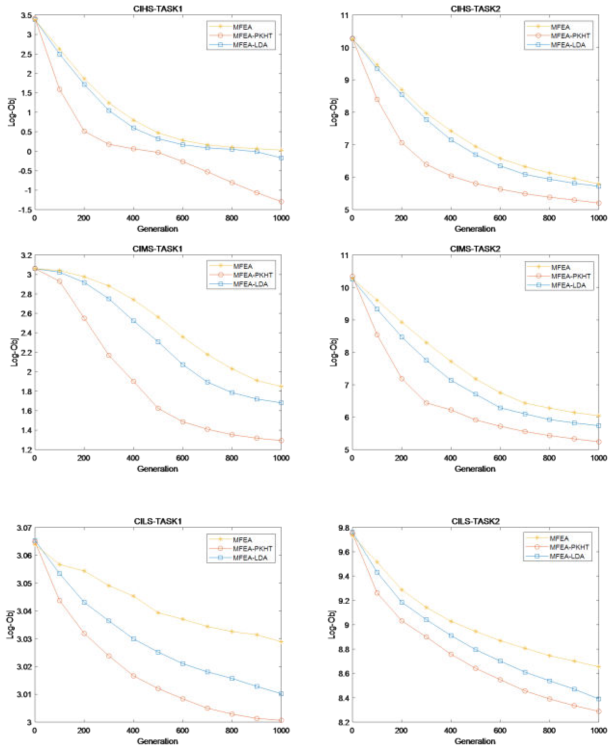
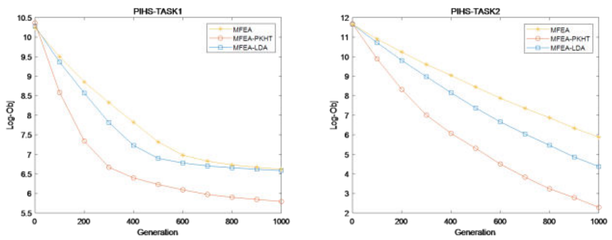


Fig. 4 Convergence curve of CIHS, CIMS, and CILS optimization tasks

Fig. 5 shows the convergence curves of MFEA-PKHT, MFEA, and MFEA-LDA on three optimization task groups: PIHS, PIMS, and PILS. It can be seen that the overall optimization performance of MFEA-PKHT in these three groups is also better than that of MFEA and MFEA-LDA, and the improvement of optimization efficiency is especially obvious in the two tasks of PILS. In this case, MFEA and MFEA-LDA produce negative migration to a certain extent as the correlation between tasks decreases, which affects the task optimization process.



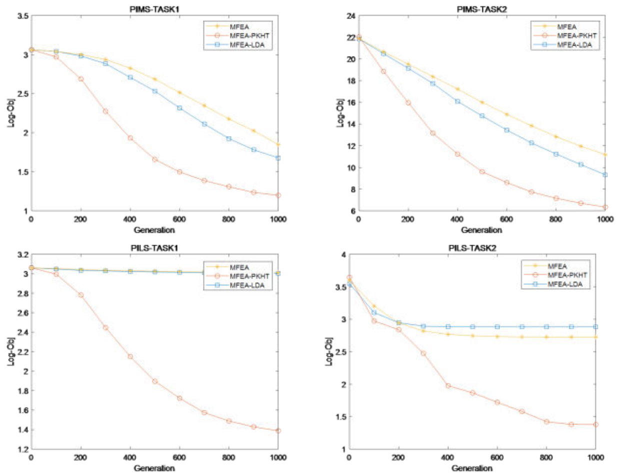


Fig.5 Convergence curve of PIHS, PIMS, and PILS optimization tasks

VI. CONCLUSION

This paper proposes a multi-task optimization algorithm MFEA-PKHT based on prior knowledge to construct helpertask tasks. Knowledge transfer in MFEA comes from multiple optimization tasks with unknown relationships, so there are some invalid or negative knowledge transfers in the process, and these non-positive knowledge transfers will reduce the optimization efficiency of the algorithm. In order to improve the algorithm's effective utilization of knowledge transfer, this paper uses a multi-task optimization algorithm to construct helpertask tasks based on prior knowledge. The helpertask tasks are constructed by decomposing the original problem into two subfunctions, which are composed of these closely related subtasks. The helpertask tasks constructed by this method are highly correlated with the original tasks, and the amount of forward knowledge transfer between these strongly correlated tasks will be greatly increased, thus speeding up the convergence speed and improving the overall optimization performance of the algorithm.

REFERENCES

[1] Mei Y, Tang K, Yao X. Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(2): 151-165.

[2] ZHU Z, WANG F, HE S. Global path planning of mobile robots using a memetic algorithm[J]. International journal of syst ems science,2013,46(11):1-12

[3] Li H H, Chen Z G, Zhan Z H. Renumber coevolutionary multiswarm particle swarm optimization for multi-objective workflow scheduling on cloud computing environment[C]//Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation. 2015: 1419-1420.

[4] Gupta A, Ong Y S, Feng L. Multifactorial evolution: Toward evolutionary multitasking[J]. IEEE Transactions on Evolutionary Computation, 2015, 20(3): 343-357.

[5] Shang Q, Huang Y, Wang Y. Solving vehicle routing problem by memetic search with evolutionary multitasking[J]. Memetic Computing, 2022, 14(1): 31-44.

[6] Feng L, Huang Y, Zhou L. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem[J]. IEEE transactions on cybernetics, 2020, 51(6): 3143-3156.

[7] Ding J , Yang C , Jin Y.Generalized Multitasking for Evolutionary Optimization of Expensive Problems[J].IEEE transactions on evolutionary computation, 2017.DOI:10.1109/TEVC.2017.2785351.

[8] Tsai C W, Lai C F, Chao H C, et al. Big data analytics: a survey[J]. Journal of Big data, 2015, 2: 1-32.

[9] Bali K K, Gupta A, Feng L. Linearized domain adaptation in evolutionary multitasking[C]//2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2017: 1295-1302.

[10] Zheng X ,Qin K A ,Gong M. Self-Regulated Evolutionary Multitask Optimization [J]. IEEE Transactions on Evolutionary Computation, 2020, 24 (1): 16-28.

[11] Han H, Bai X, Han H. Self-adjusting multitask particle swarm optimization [J].IEEE Transactions on Evolutionary Computation, 2021, 26(1): 145-158.

[12] Shi L, Hu Z, Su Q. A modified multifactorial differential evolution algorithm with optima-based transformation[J]. Applied Intelligence, 2023, 53(3): 2989-3001.

[13] Wu D, Tan X. Multitasking genetic algorithm (MTGA) for fuzzy system optimization[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(6): 1050-1061.

[14] Knowles J D, Watson R A, Corne D W. Reducing local optima in single-objective problems by multi-objectivization[C]//International conference on evolutionary multi-criterion optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001: 269-283.

[15] Lochtefeld D F, Ciarallo F W. Multi-objectivization via decomposition: An analysis of helper-objectives and complete decomposition[J]. European Journal of Operational Research, 2015, 243(2): 395-404.

[16] Handl J, Lovell S C, Knowles J. Multiobjectivization by decomposition of scalar cost functions[C]//International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008: 31-40.

[17] Da B, Gupta A, Ong Y S. Evolutionary multitasking across single and multi-objective formulations for improved problem solving[C]//2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2016: 1695-1701.

[18] Rawlins G J E. Pareto Optimality, GA-easiness and Deception[C]//Genetic Algorithms. Morgan Kaufmann Publishers, 1993: 118.