Piggybacking on Past Problem for Faster Optimization in Aluminum Electrolysis Process Design

Lizhong Yao[a], Tiantian He[b], Haijun Luo[a,*]

[a]*College of Physics and Electronic Engineering, Chongqing Normal University, Chongqing , PR China*
[b] *Institute of High Performance Computing, Agency for Science, Technology and Research (A*STAR),Singapore*

Abstract

In process industry, it is a habitual phenomenon that the production efficiency is improved via replacing the obsolete devices with the advanced ones. To achieve an optimal performance, these devices are always required for heavily empirical adjustments, which is very time-consuming and inefficient. Though outdated, invaluable experiences for adjusting devices towards a more efficient industrial production have been accumulated by those replaced facilities. Thereby, new devices are expected to adapt to perform the industrial task without many modulations if the aforementioned experiences can be appropriately utilized. Inspired by the fact that evolutionary multitasking is capable of exploiting latent similarities and commonalities among multiple optimization tasks so as to improve the overall convergence of multi-task optimization, in this paper, we propose a novel framework to automatically search for the optimal settings for new devices based on the knowledge accumulated by the old. The framework, dubbed Piggybacking on Past Problem for Faster Optimization (PPPFO), is able to piggyback on the past optimization problem for a faster convergence of the targeted. By means of automatically transferring search experiences (i.e. genetic and cultural characteristics) from source task to the target, PPPFO can assist an engineering optimizer to improve its search exercises. PPPFO has been tested with a number of widely used benchmark functions and has been successfully adopted to an important real application, i.e., aluminum electrolysis process design. The remarkable results verify the efficacy and efficiency of the proposed framework.

Keywords: Aluminum electrolysis, Evolutionary multitasking, Multi-objective optimization

1. Introduction

Metallurgical industries like aluminum electrolysis are always huge energy consumers around the world (Haraldsson and Johansson, 2018, Li et al., 2021). Thereby, economizing on the energy consumption simultaneously preserving both production efficiency and quality has drawn much interest in the recent research (Ding et al., 2021, Yao et al., 2019). To make an advanced electrolyzer, it has to satisfy two contradictory requirements in the meantime, which are maximizing the current efficiency and minimizing the energy consumption (Yi et al., 2017). In general, these conflicting goals can be simultaneously achieved via two ways. One is to use advanced optimization control theory, e.g., population-based search techniques (Yi et al., 2017, Yao et al., 2022) to simultaneously optimize related indicators concerned by aluminum electrolysis. The other is to update the hardware

*corresponding author
Email addresses: 20220009@cqnu.edu.cn (Lizhong Yao), tiantian.he@outlook.com (Tiantian He), Luohaijun@cqnu.edu.cn (Haijun Luo)

of the electrolytic cell, such as ameliorating cathode structure (shaped cathode) (Li et al., 2012) and anode structure (perforated anode) (Tian, 2011).

Without loss of generality, the aforementioned first way to improve the production process of aluminum electrolysis can be formulated as a multi-objective optimization problem, which aims at optimizing production efficiency and energy consumption at the same time. As these two objectives are mutually competitive in nature, an optimal solution to the design parameters in an aluminum electrolysis equipment should be optimal trade-offs of design parameters (Wang et al., 2021a, Shang et al., 2021), which takes into consideration both objectives. Given these properties, a Pareto front (PF) can be obtained when a multi-objective optimization problem is solved (Gong et al., 2018, Deb et al., 2019, Li et al., 2020b). Empirically, such problems can be resolved via population-based searching algorithms, e.g., evolutionary optimization, and there have been several effective techniques proposed in the recent. For example, Yi et al. (2017) proposed an improved multi-objective quantum-behaved particle swarm optimization (IMQPSO) algorithm to optimize operating parameters of aluminum electrolysis equipment. In Yi et al. (2018), the ar-MOEA was devised for evolutionary multi-objective optimization in the aluminum electrolysis process design. Besides, other approaches to evolutionary optimization, including NSGA-II (Deb et al., 2002), SPEA/2 (Zitzler et al., 2001), IBEA (Zitzler and Künzli, 2004), SAEA (Chugh et al., 2016), MOEA/D (Zhang and Hui, 2007), RMOEA-DVA (Liu et al., 2022a) and B-SAEA (Liu et al., 2022b) , have also been widely applied to industry-related optimization and obtained satisfactory performance in the real world. Though effective to some extent, the accumulated optimization experiences, which are obtained from previous optimization process (Yi et al., 2017), are not fully utilized or even ignored.

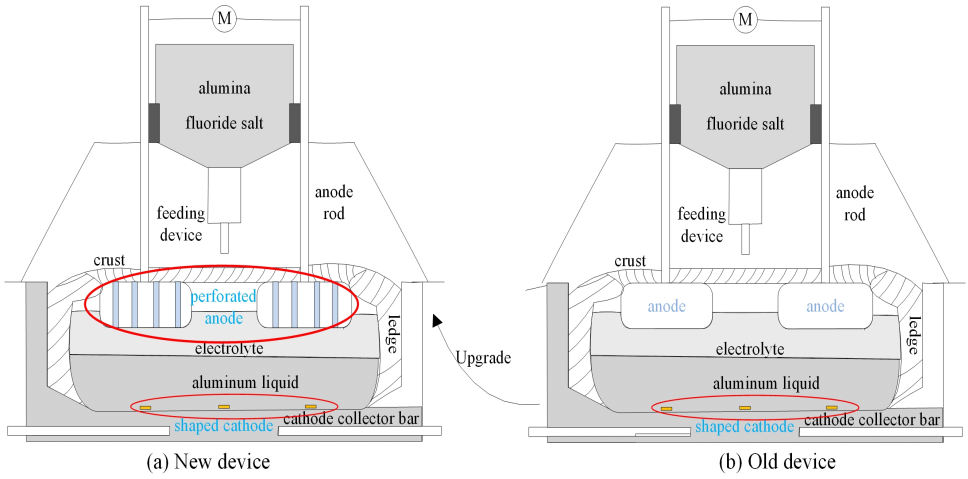

Fig. 1. An updated behaviour from old device to the new

Compared with improving the production performance using above evolutionary algorithms, updating the electrolytic hardware to advance the aluminum electrolysis is sometimes more challenging. It is well known that the domain, as well as the decision parameters of the devices are changed along with the update of production process (Hou et al., 2020). As a result, the best combination of design parameters has to be re-explored to ensure the updated device can be optimally operated. Let us take a typical updated behavior of the aluminum electrolysis device as an example, which is depicted in Fig. 1. As the figure shows, except for jointly using shaped cathode and perforated anode, the rest of the two devices are almost same. Though such a little alteration, the engineers always have to try their best to adapt the updated device to obtain its optimal performance. In the real aluminum electrolysis process, such adaptions have always to be manually performed (Gui et al., 2018)

and cost several months or even longer. Given this deficiency in modern aluminum electrolysis, we have the following findings that motivate us to propose a more efficient and effective approach to shorten the time of adapting new electrolysis devices to the optimal production process. The renewal of aluminum cell is always incremental, i.e., only few key components are replaced with the more advanced, while the most remain same. Therefore, it makes sense that the optimal combinations of design parameters of old devices are somewhat helpful to seek for the optimal settings of the updated device.

Experience is often the best teacher (Gupta et al., 2018). Assuming the old device plays the role of a teacher, it would be significantly attractive if we could develop some rules that can automatically transfer the existing optimization experiences to the new and related optimization tasks (as students) (Min et al., 2017, Yi et al., 2020), so as to piggyback on past problem (old device) for a faster optimization on the new device (PPPFO). Inspired by Multi-Objective Multifactorial Evolutionary Algorithm (MO-MFEA) (Gupta et al., 2017, Bali et al., 2020), in this paper, we propose a novel optimization framework aiming at fulfilling the proposition of PPPFO. What makes the proposed framework more advanced is that PPPFO can exploit latent similarities and commonalities between source and targeted tasks. And then the overall performance of optimized task is improved by means of automatically transferring search experiences (i.e. genetic and cultural characteristics) from source task to the target. The proposed optimization framework considers the following four aspects to improve the optimization performance of the updated device in aluminum electrolysis process, which can be seen as the contributions of this paper.

- Through analyzing the relationship between old and new optimization tasks in aluminum electrolysis, we design an efficient model for device renewal, which is capable of utilizing rich experiences accumulated by previous task of optimization.
- To ensure the cumulative experiences to be fully utilized by the targeted optimization task, we propose a random-keys-inspired mapping link to encode domains of source and targeted optimization tasks into a common search space where beneficial experiences are allowed to be transferred efficiently.
- By integrating the aforementioned two novel ingredients with MO-MFEA, we propose a novel framework, dubbed PPPFO, to piggyback the past optimization problem for a faster convergence of the targeted.
- The proposed PPPFO has been tested with a number of benchmark functions, and applied in a real industrial optimization problem, i.e., aluminum electrolytic process design. The experimental results show that PPPFO can significantly improve the overall performance of optimization problems.

The rest of this paper is organized as follows. In Section 2, we briefly investigate the related works. In Section 3, we elaborate the proposed framework of PPPFO. In Section 4, the proposed framework is extensively tested with a number of benchmarks. A case study which shows the proposed PPPFO is successfully used to accelerate the process design of aluminum electrolytic is presented in Section 5. A summary is provided in Section 6.

2. Related works and motivations

2.1. Related works

Inspired by NSGA-II and the bio-cultural models of multifactorial inheritance which are helpful in exploring transfer mechanism of pertinent information in multi-tasking environment (Gupta and

Ong, 2016), MFEA is firstly proposed in (Gupta et al., 2016) which is based on the assumption that a specific connection exists between the source task and the target task for knowledge transfer. However, once the assumption does not hold, it may lead to a degraded performance of MFEA, even inferior to that of the unmigrated single-task algorithm. In recent years, to address the above-mentioned problems in knowledge transfer, many studies have been proposed to improve the performance of MFEA. EMT-EGT (Feng et al., 2019) was proposed to use noise reduction autoencoders to learn linear transformations between subpopulations to transfer knowledge. Similarly, LDA-MFEA (Bali et al., 2017) implemented knowledge transfer by transforming the search space of the task. MFEA-II (Bali et al., 2020) adaptively updated the random mating probability based on online learning and task similarity. In addition, many other studies, such as G-MFEA (Ding et al., 2019), MFEA-GHS (Liang et al., 2019), MTO-DRA (Gong et al., 2019), MFMP (Li et al., 2020a), and MTEA-AD (Wang et al., 2022), are successively proposed to solve multi-task optimization problems.

To further enhance the extensibility of MFEA, MO-MFEA was proposed for multi-objective optimization problems in a multi-factorial evolutionary environment (Gupta et al., 2017). For improving the performance of the multi-objective multi-task algorithm, Bali et al. (2021) used the probability distribution to model the target population and analyze the impact of inter-task knowledge interactions. Liang et al. (2022) improved the quality of inter-task knowledge transfer by using linear transformations to map the population to a low-dimensional subspace. Yi et al. (2022) proposed MFEA/IDD that combines the advantages of both multitasking evolution and decomposition-based evolution strategies to balance the convergence and diversity among multiple objectives. Furthermore, multi-factorial evolutionary methods by integrating different heuristic strategies, i.e., DAVT (Gao et al., 2022), meme helper-tasks (Ma et al., 2022), particle swarm (Ji et al., 2023), and graph-based hyper-heuristics (Hao et al., 2021), have been proposed in different application areas. In summary, the above studies are gradually developed under the inspiration of MO-MFEA framework.

2.2. Research motivations

The main difference between methods based on MO-MFEA and NSGA-II is MO-MFEA is able to simultaneously solve multiple problems by leveraging the implicit parallelism of population-based search to exploit potential complementarity among multiple problems. In contrast, NSGA-II concentrates to tackle only a single optimization problem at a time for getting optimal trade-off of different objectives. Due to its universality, MFEA has been successfully used to solve various optimization problems (Gupta et al., 2017, Bali et al., 2020, Wang et al., 2021b, Yi et al., 2022, Feng et al., 2021), and has attracted growing research interests (Binh et al., 2021, Liu et al., 2018, Martinez et al., 2022).

Without loss of generality, in the algorithm of MO-MFEA, we assume that K distinct optimization tasks all are minimization problems. The MO-MFEA defines a unified search space Y that encompasses X_1, X_2, $\cdots$, X_K, where X_i (i=1, 2, $\cdots$, K) represents the i^{th} design space in the i^{th} task. So, every individual p_j (j=1, 2, $\cdots$, P, where P is the population size) can be encoded into the unified search space (Y). Based on the above background, some technical terms that are commonly encountered in the MO-MFEA are listed below.

Definition 1 (Skill Factor): In a multi-tasking evolutionary environment, the skill factor τ_j attributed to an individual p_j is the task including individual p_j among all the tasks. So, the skill factor τ_j belongs to the interval [1, K], where K is the number of tasks.

Definition 2 (Scalar Fitness): The scalar fitness φ_j of individual p_j can be calculated by the following formula $\varphi_j = 1/r^j_{\tau_j}$, where $r^j_{\tau_j}$ is the rank index of p_j in the task τ_j sorted by descending order preference. So, the scalar fitness φ_j belongs to the interval (0, 1].

The scalar fitness is helpful to compare performance with the different individuals in a simplistic way. The skill factor is a very important and meaningful parameter, because it represents the cultural

identity or cultural background of an individual. Based on the principle of cultural transmission assortative mating (Cloninger et al., 1979a,b, Rice et al., 1978), it can be known that those individuals who have matching cultural traits prefer to mate with each other freely. In contrast, other individuals with cross-cultural traits may crossover following a mating probability. In other words, the core parameters like the skill factor and the mating probability, in the MO-MFEA, determine the strength and direction of genetic crossover and mutation. For the sake of brevity, the reader can get more details about the mating probability by referring to Gupta et al. (2017).

Inspired by the above theory, optimizing new device in aluminum electrolysis and other industrial processes while concerning the optimization experiences of the old are just right to be viewed as a multifactorial multitasking problem. So, MO-MFEA has great potential to help solve such a proposition like PPPFO. Though PPPFO is developed based on MO-MFEA, the proposed PPPFO is still distinct from original MO-MFEA due to the uniqueness of the optimization problems solved by the proposed framework. (1) In the MO-MFEA, the different tasks play the roles of student and student in the multi-tasking optimization. In contrast, the old device and new device in the PPPFO play the roles of teacher and student. Taking into account the distinction, we must pay more attention to the following point. Different production models should be built to characterize new and old models separately as the multifactorial objective functions. Especially, the old model is a representative with rich design experience. (2) For PPPFO, the migration of design space is commonplace with the upgrading of old device. A fair mapping link must be designed when design experience is transferred from the old device to the new one. Random keys (Gonçalves and Resende, 2011) often viewed as a chromosome will be introduced to implement the mapping transformation for the above problem. If the design space of decision parameters is regarded as the phenotype, then the random keys are the genotype of the design space.

Based on these inspirations and the aforementioned practical motivation, we aim to propose a novel framework of piggybacking on past problem for faster optimization (PPPFO) with the notion of MO-MFEA and apply this framework to real industrial process in this paper.

3. The optimization framework PPPFO

3.1. The countermeasures for distinctions

Generally speaking, PPPFO is a problem-solving framework in which MO-MFEA is only an ingredient. In order to facilitate interpretation and comparison between them, in this paper, we call this framework PPPFO based on MO-MFEA as PPPFO. Next, the countermeasures for distinctions will be introduced.

3.1.1. Countermeasure for different roles between tasks or problems

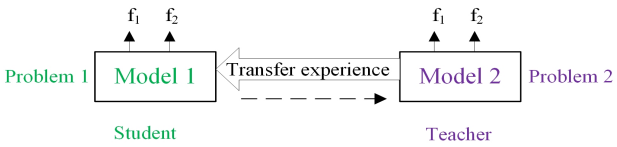

Fig. 2. The roles of different problems in PPPFO

In our case study as shown in Fig. 1, the old device is the embodiment of mainstream equipment of aluminum electrolysis and currently applied on a large scale at factory. So, the old device has accumulated a wealth of valuable design experience and will be treated as a teacher in this paper.

Fig. 3. The roles of different problems in MO-MFEA

However, the new device is currently under the experimental production stage and will be treated as a student correspondingly. This feature of PPPFO is a relationship between teacher and student as shown in Fig. 2. Meanwhile, it can be seen that the Model 2 (like a teacher) does not care or might ignore those transfer experience from model 1 (like a student), because the experience is quite immature and often no practical value. In contrast, model 1 hopes to absorb more excellent transfer experience from model 2. In the MO-MFEA, the role of each task or problem in the multifactorial environment is completely consistent with each other and all will be treated as students. Therefore, the feature of MO-MFEA is the relationship between student and student as shown in Fig. 3. It indicates that model 1 and model 2 (all like students) hope to absorb transfer experience from each other. So, about the role, PPPFO and MO-MFEA are fundamentally different.

Facing the distinction mentioned above, the following three steps are proposed as the countermeasures. Firstly, the corresponding design parameters which need to absorb transfer experience must be selected for determining design space. Then, it needs to be determined which is the past problem as a teacher and which is the new problem as a student in the production process. It is not difficult to complete the above two steps. Lastly, mathematical models of the old and new problems must be established as the mathematical descriptions of different problems or tasks. Theoretically, feed-forward neural networks can approximate any nonlinear function. Taking into account our problem characteristics and industrial devices, the neural networks are adopted to build mathematical models of different devices. Here, a basic structure of neural networks with multiple inputs and 2 outputs is shown below in Fig. 4.

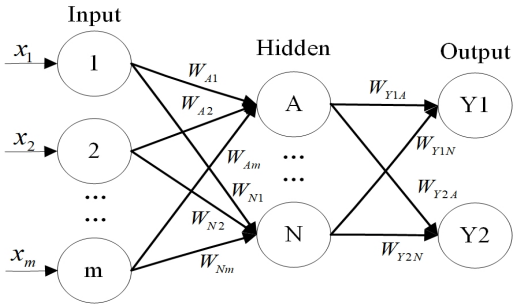

Fig. 4. A basic structure of neural networks with multiple inputs and 2 outputs

The main model parameters are as follows. The detailed algorithm flow will be presented in Section 3.2.1.

a. connection weight W_{ij}, $i = 1, \cdots, N; j = 1, 2, \cdots, m$ indicates that it is the weight between hidden layer neuron (i) and input neuron (j), where, N is the num of hidden layer neurons, m is the num of output neurons.

b. connection weight W_{Yki} (k=1, 2) indicates it is the weight between output neuron (Yk) and hidden layer neuron (i).

c. The logistic sigmoid function $g(x)$ is used as the activation function of the hidden layer. It

is given by the following expression, where, x is the input value of the corresponding hidden layer neuron.

$$g(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

d. The transfer function of the output layer uses a linear function, i.e., purlin function. So, the final output of neural networks can be obtained as follows, where, Yk is the output value of the output neuron Yk.

$$Y_k = \sum_i W_{Yki} g\left(\sum_j W_{ij} \cdot x_j\right),\ k = 1,\ 2 \tag{2}$$

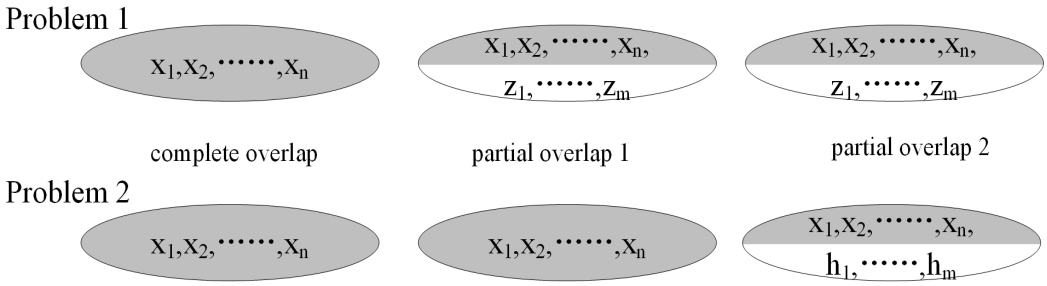

Fig. 5. Three overlapping patterns in design space,where x, z and h are different design parameters.

3.1.2. Countermeasure for the migration of design space

This phenomenon is very commonplace that an upgrade of the old device which usually breaks the original production balance will inevitably need a new production process to match it. In other words, new design parameters must be exploited to match new process. Even if the new design parameter set may be complete overlap or partial overlap compared with the past as shown in Fig. 5, where overlapping design variables are filled with a gray background, the design ranges of these parameters will also generally change. In our case of aluminum electrolysis process, the design space is complete overlap. So, the design parameters of new and past problems both are X, where, Xnew= (x_1, x_2, $\cdots$, x_m), Xpast= (x_1, x_2, $\cdots$, x_m). Then, we assume that the design ranges (DR) of different parameters in the Xpast and Xnew are given as follows.

$$\text{DRpast} = \begin{bmatrix} lower_1^p, upper_1^p \\ lower_2^p, upper_2^p \\ \vdots \quad\quad \vdots \\ lower_m^p, upper_m^p \end{bmatrix} \tag{3}$$

$$\text{DRnew} = \begin{bmatrix} lower_1^n, upper_1^n \\ lower_2^n, upper_2^n \\ \vdots \quad\quad \vdots \\ lower_m^n, upper_m^n \end{bmatrix} \tag{4}$$

where, $(lower_i{}^p, upper_i{}^p)$ and $(lower_i{}^n, upper_i{}^n)$ represent respectively the design ranges of the i^{th} parameter in the Xpast and Xnew; the superscripts p and n represent the past problem and new problem, respectively.

Considering the distinction between the DRpast and DRnew, it's not a wise move to transfer the experience directly from the past problem to the new one. If doing that directly, it may go beyond the design range of new problem or pass unharmonized design experience to the new one, which might lead to some meaningless results. Based on the above analysis, a reasonable mapping relationship needs to be established between DRpast and DRnew for bridging the gap. In this paper, a mapping method based on random keys is proposed to solve this problem from a practical standpoint. The random keys (Andrade et al., 2021) represent essentially the genetic material of different individuals which can be shared and transferred among different problems. More precisely, each random key represents a design variable. If the design variable itself is viewed as the phenotype, then the random key is its genotype. So, we need to transform the design space of the past problem into a random key space, wherein each random key is assigned between 0 and 1. Then, use random keys as a link to map it to the design space of the new problem.

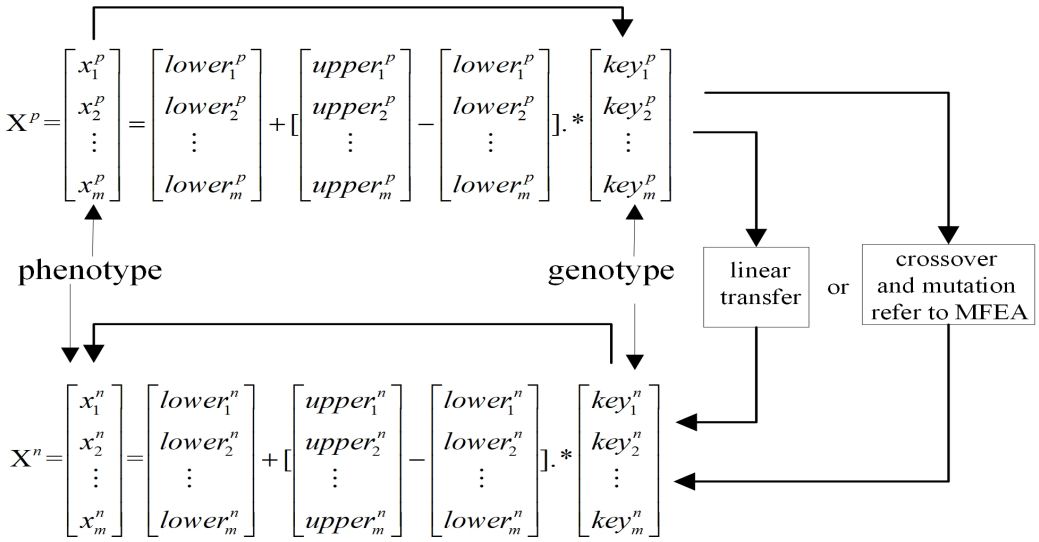

Fig. 6. A mapping process from DRpast to DRnew

A simple mapping process from X^p as a design space of past problem to X^n as a design space of new problem is shown in Fig. 6, where, the key_i^p(i=1, 2, ⋯, m) can be obtained by the mapping process from X^p to key_i^p based on $lower_i^p$ and $upper_i^p$; the key_i^n (i=1, 2, ⋯, m) can be got based on key_i^p by a way of transfer experience. Then, a phenotype of X^n can be calculated by inverse mapping based on key_i^n. So, the transfer experience will be carried out between random keys. The random keys key_i^p and key_i^n (i=1, 2, ⋯, m) become the link connecting X^p and X^n. In Fig. 6, the linear transfer denotes a transfer or injection way that key_i^n is equal to key_i^p, and related operators and experiments will be presented in Section 5. The crossover and mutation involving the skill factor and mating probability based on MO-MFEA will be listed in the next section. In short, through the analysis and discussion of distinctions between past and new problems, a novel framework of piggybacking on past problem for faster optimization (PPPFO) is formulated as shown in Fig. 7. The details of the framework will be presented next.

3.2. Ingredients of the framework PPPFO

Fig. 7 describes the whole framework and ingredients about the proposition of piggybacking on past problem for faster optimization (PPPFO). In general, the PPPFO includes the following ingredients, including variable selection, building models, single problem optimization based on

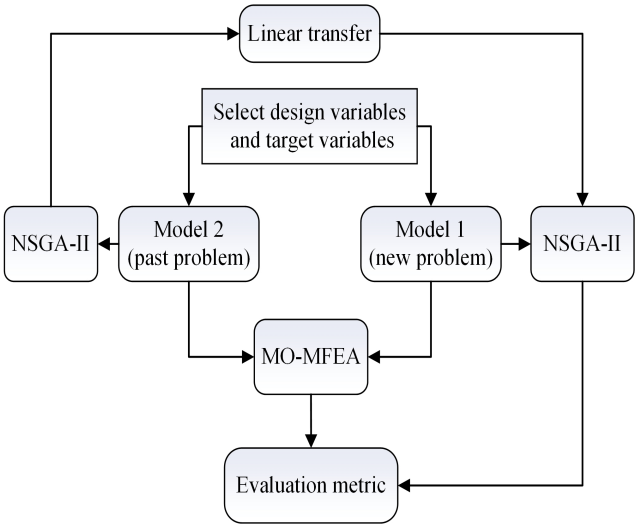

Fig. 7. The framework of PPPFO

Algorithm 1: Basic structure of PPPFO

```
Select design variables X and target variables F (f1, f2 in our case) of different problems;
Identify past and new problems, get X^p, DRpast, Fpast and X^n, DRnew, Fnew;
Establish mathematical models for the characterization of past problem (model 2) and new
  problem (model 1) (see Algorithm 2);
Use NSGA-II to solve PF of different models while saving the solutions of each generation;
Transfer process (see Algorithm 5);
Implement MO-MFEA (see Algorithm 6);
Evaluate performance indicators
if real PF is known then
    Evaluation metric uses Inverted Generational Distance (IGD);
    else if real PF is unknown then
        Evaluation metric uses Hypervolume (HV);
    end
end
```

NSGA-II, linear transfer, 2-factorial problems optimization based on MO-MFEA and evaluation metrics. Thereafter, in order to show the process in more detail, we present the procedure of PPPFO in Algorithm 1.

The PPPFO begins by selecting design variables X and target variables F of different problems. Strictly speaking, the PPPFO belongs to the 2-factorial evolutionary environment or 2 distinct optimization problems that denote the past problem and new problem, respectively. Based on the basic concepts, the design variables include Xpast and Xnew following the target variables Fpast and Fnew. Meanwhile, the design ranges of DRpast and DRnew cannot also be neglected. When doing all the above processes, we have unconsciously defined the role who is the past problem (like a teacher) as the Mode 2 or the new problem (like a student) as the Model 1. Then, we will establish mathematical models for characterizing past problem (Model 2) and new problem (Model 1).

Algorithm 2: Modeling process based on neural networks

Split the data sample into a training set and a test set;
Data normalization;
Set the number of hidden layer neurons (see Algorithm 3);
Build BP neural networks;
Set network parameters (number of iterations, learning rate and training goal);
Train neural networksm;
Model prediction and denormalization
Save net (including W_{ij} and W_{Yki}) of mode 1 and model 2 respectively.

Algorithm 3: Select the optimal number of hidden layer neurons

for *run 30 times* **do**
    for *n from* $\sqrt{m+o}+1$ *to* $\sqrt{m+o}+10$ **do**
        Implement Algorithm 2;
        Calculate the mean square error (MSE) and save;
    end
end
Select the optimal number of hidden layer neurons based on MSE as the setting value in Algorithm 2;

Algorithm 4: Select the best net

for *run 30 times* **do**
    Use the optimal number of hidden layer neurons from Algorithm 3;
    Implement Algorithm 2;
    Calculate the perdition errors and save them every run;
end
Select the best net based on perdition errors and save net;

3.2.1. Modeling of past and new problems

Detailed modeling process can be found in Algorithm 2. Both Model 1 and Model 2 will execute the algorithm independently. In the algorithm, it is a well-worn question on how to select the number of hidden layer neurons (n). The conventional practice is still to use the trial and error method based on the following formula.

$$n = \sqrt{m+o}+l \tag{5}$$

where m is the number of input neurons, o is the number of the output neurons and l is a constant between 1 and 10.

In order to offset the impact of the uncertainty from the initial value of the neural networks, we propose using the following strategies (see Algorithms 3 and 4) to obtain the optimal number of hidden layer neurons and the optimal network structure.

After completing all the above procedures, two separate models will be got and then be served as the 2-factorial problems for subsequent optimization.

Algorithm 5: Patterns of transfer experience

1 Two patterns of transfer experience can be implemented as shown in Fig.7;
2 Linear transfer: Perform linear transfer. So, $key_i{}^n = key_i{}^p$;
3 Nonlinear transfer: Perform genetic operator like MO-MFEA, (see Algorithm 6 and 7).

3.2.2. Multi-objective optimization based on NSGA-II

NSGA-II (Deb et al., 2002), one of the most popular multi-objective optimization algorithms, is called nondominated sorting genetic algorithm with elitist strategy. It excels at solving an optimization problem at a time. So, in the framework of PPPFO, we use it to separately solve and approximate the PF of the Model 1 and Model 2 with saving the solutions of each generation. There are two purposes for doing this. One is to test whether model 2 (like a teacher) is better than model 1(like a student), because we hope that model 2 can represent a good design experience. The other is to record and save some design experience of optimization process from past problem for the use to cooperate with linear transfer in Algorithm 5.

3.2.3. Pattern transfer

We have already introduced the transfer of design space in detail in Section 3.1.2. It is a fairer way to complete the transfer experience in the genotype space as described with random-keys (from $key_i{}^p$ to $key_i{}^n$). In the framework of PPPFO, two patterns of transfer experience are available as shown in Algorithm 5. When we perform linear pattern, the liner transformation will be carried out. In the pattern, the design experience from the past problem (Model 2) that we have saved when using NSGA-II to approximate its PF will be transferred or injected to the new problem (model 1) directly. The specific injection ways and operations will be presented in Section 5. If the nonlinear pattern is implemented, a nonlinear transformation based on SBX crossover and polynomial mutation in 2-factorial problems will be employed to transfer design experience from the past problem to the new one. In other words, the pattern is a genetic operator of evolutionary computation in common with MFEA including genetic and biocultural characteristics (see Algorithms 6 and 7).

3.2.4. Multi-objective multifactorial evolutionary algorithm (MO-MFEA)

MO-MFEA presented in Algorithm 6, an ingredient in the framework of PPPFO, plays an important role in piggybacking the past problem for faster optimization. In order to achieve effective transfer experience to the new problem, we must clearly explain and discuss the following three important parameters, i.e., scalar fitness based on preference ordering, skill factor and random mating probability, and one evaluation indicator.

Scalar fitness φ can be calculated based on NF and CD. To help understand the concept better, we describe the calculation process of scalar fitness as shown in Fig. 8. For a 2-objective problem with the population P= $p_1, p_2, \cdots, p_9$, we first need to get the nondominant relationship of all the individuals based on Non-dominated Front (NF), for example $1^{st}\text{PF} < 2^{ed}\text{PF} < 3^{rd}\text{PF}$ (as illustrated in the Fig. 8). Then, calculate CD of each individual in different PFs. For instance, in the 1^{st}PF, p_1, p_2 and p_4 can be sorted based on CD, where the larger the CD is, the better priority the individual has. So, we can get the sort of all individuals based on descending preference ordering r^j ($j = 1, 2, \cdots, 9$). The scalar fitness can be calculated by $\varphi_j = 1/r^j$. The greater the scalar fitness is, the better the individual becomes. Therefore, using the scalar fitness is very convenient to judge which individual is better or worse.

Skill factor essentially represents the cultural identity or cultural background among different individuals. The cultural transmission and genetics will all be viewed as types of inheritance and

Algorithm 6: Basic procedure of MO-MFEA

Generate an initial population P_0 with N individuals;
Set the skill factor τ_j of every individual and evaluate the individual;
if τ_j *belongs to the past problem (model 2)* **then**
    Use the model 2 to evaluate it;
    else if τ_j *belongs to the new problem (model 1)* **then**
        Use the new problem (model 1) to evaluate it;
    end
end
Use the Non-dominated Front (NF) and Crowding Distance (CD) to compute the fitness of each individual;
Initialize gen=0;
while *the termination condition has not been met* **do**
    Implement a binary tournament to choose a new population P_{gen} from P_0;
    Carry out genetic operator, get population goPgen (see Algorithm 7);
    Get an intermediate population by merging populations as follows, imPgen=[goPgen; Pgen];
    Update the scalar fitness of each individual based on NF and CD;
    Extract N individuals with optimum fitness from imPgen as the next generation of population P_{gen+1};
    gen=gen+1;
end

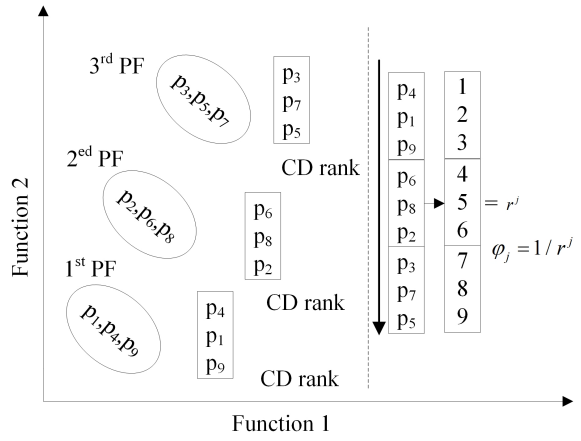

Fig. 8. The calculation process of scalar fitness for a specific problem

co-create the phenotype of offspring. In the present framework, the skill factor of every problem will be assigned randomly. It is important to note that once a problem with a population is assigned a skill factor, all individuals of the population will carry the skill factor. Doing this will ensure that each individual can be evaluated with respect to merely one problem. In the paradigm of PPPFO, the skill factor can only be 1 and 2, because the framework belongs to the 2-factorial evolutionary environment including only past problem (Model 2) and new problem (Model 1). To elaborate, if the skill factor of new problem is set to 1, the model 1 will be used to evaluate individuals who are carrying the same skill factor (see Algorithm 6).

Algorithm 7: Genetic operators in multifactorial environment

1 Suppose that p_i and p_j are randomly selected from population P_{gen} as the candidate parents. p_i and p_j are essentially chromosomes with random keys like $key_i{}^p$ and $key_i{}^n$;
2 Set the random mating probability by online learning;
3 Implement SBX crossover and polynomial mutation, get $p_i^{'}$ and $p_j^{'}$;
4 Randomly assign new skill factor τ_i and τ_j to $p_i^{'}$ and $p_j^{'}$;
5 evaluate $p_i^{'}$ and $p_j^{'}$ based on their corresponding τ only;
6 Get offspring goPgen after all individuals have completed the genetic operators

Random mating probability (Rmp) and skill factor determine the implicit transfer of chromosomes among candidate solutions, because the two parameters can regulate the 'transfer experience' existing in the gene-culture type of inheritance. In the framework of PPPFO, the implicit transfer is achieved by Algorithm 7. Rmp reflects the mindset of decision makers to control the intensity of experience exchange between different problems. Generally speaking, the Rmp can be set between 0 and 1 or be got by online learning (Bali et al., 2020). If the two problems are closely related, then it is best to set it close to 1 or online learning. In the PPPFO, the aim is to transfer experience or boost free exchange of genetic material from the past problem to the new as much as possible. So, the Rmp should be set to online learning in PPPFO. Then, all the individuals are implemented by SBX crossover and polynomial mutation with no barriers. With regard to skill factor, each individual can select a skill factor randomly to imitate one of the parent for inheriting parental cultural characteristics. Based on the above analysis, the new variation of the genetic operator is presented in Algorithm 7.

Evaluation Metrics. In order to effectively compare the performance between PPPFO and NSGA-II, the metrics of Inverted Generational Distance (IGD) and Hypervolume (HV) (Jiang et al., 2014) will be employed. The two metrics both can present a reasonable and comprehensive assessment of not only the convergence but also the diversity to the approximate PF. However, when the true pareto front is available (as is the case of benchmark functions), the IGD should be adopted to evaluate the performance between the true PF and the approximate PF, because Euclidean distance of each iteration between the two PFs can be computed by using IGD. So, the smaller the IGD value is, the better the performance of approximate PF gets. When the true PF is unknown (as is the case of complex industrial process in real problems), the HV metric (known as Lebesgue measure) is more suitable to be used to evaluate the approximation performance of pareto solutions, where the reference point should be set away from the ideal PF. Compared with IGD, the larger its value is, the better the approximation performance becomes. This difference is vividly described in Fig. 9. For the sake of fairness, the averages of IGD and HV should be calculated based on dozens of independent runs.

Computational Complexity of the proposed framework. As for the cost of accumulating experience, such knowledge accumulation does not require additional computational cost, as the source and target tasks are essentially separated (but possibly similar) and the knowledge from source tasks is readily available when a new optimization task is performed. The cornerstone of the PPPFO is MO-MFEA, which is developed based on NSGA-II. The overall complexity of the NSGA-II is $O(MN^2)$, which had been analyzed in detail in the literature (Deb et al., 2002). Where, M is the number of objective functions and N is the population size. Considering the MO-MFEA is developed based on NSGA-II, the main operations of MO-MFEA different from NSGA-II are presented in the

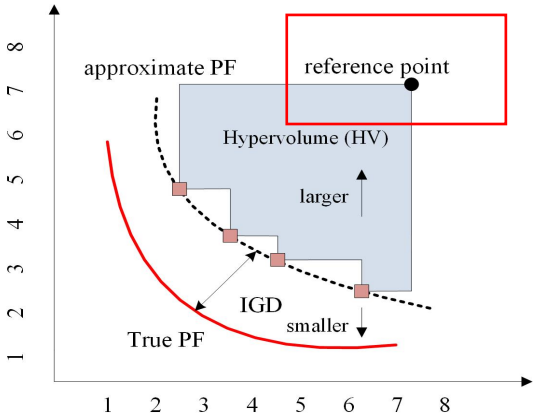

Fig. 9. Metrics of IGD and HV

following steps of Algorithm 6. (1) step 2 requires $O(N)$ to set the skill factor; (2) step 9 computes the fitness of each individual with worst-case complexities including Non-dominated Front ($O(MN^2)$), Crowding Distance ($O(MNlog(N))$) and Scalar fitness with sorting ($O(Nlog(N)+N)$); (3) step 13 still needs additional computing sill factor $O(N)$; (4) step 15 updates the scalar fitness requiring $O(M(2N)^2)$, because the intermediate population is a combination population with $2N$ individuals. Therefore, the overall complexity of the optimized framework is $O(GMN^2)$, where G is the number of generations. So, it is obvious that there is no significant difference in computational complexity between MO-MFEA and NSGA-II and no additional cost is required by PPPFO.

4. The effect of PPPFO based on benchmark functions

In this section, we extensively test the effectiveness of PPPFO using synthetic benchmark functions as introduced in reference (Gupta et al., 2017). When testing PPPFO using benchmark functions, the Model 1 (new problem like a student) and Model 2 (past problem like a teacher) as shown in Fig. 7 should be replaced by corresponding functions. This is different from the evaluation of PPPFO using real industrial problems, where a device or process with sufficient design experience will be first used as Model 2 and new device or process with lack of design experience will be selected as the Model 1. Obviously, selecting different benchmark functions for Models 1 and 2 may significantly influence the overall performance of PPPFO. To allow PPPFO to be appropriately tested, we select the benchmark functions for PPPFO as teachers and students according to the following scheme. The approximation performance (in terms of IGD value) of each benchmark function is firstly tested by NSGA-II. Those functions with better approximation performance (a smaller IGD value) are selected for Model 2. Consequently, the rest functions, which are hard to approximate in general, are used in Mode 1. Since the true PFs of benchmark functions are known, the IGD indicator are employed to showcase the performance between PPPFO (handling two problems synchronously) and NSGA-II (handling only a problem at a time).

4.1. Benchmark functions

In Table. 1, the ZDT4-R/G/A/RC all are the variants based on ZDT4 simply. ZDT4-R/G/A can be got simply by changing the multimodal function g(x). ZDT4-RC is a constrained variation based on ZDT4-R. The four benchmarks represent four single problems with 2 objective functions in each problem. It should be noted that (ZDT4-R, ZDT4-G) indicates a combination of ZDT4-R and ZDT4-G, which is viewed as 2-factorial problems in the PPPFO. (ZDT4-A, ZDT4-RC) is another a pair

Table 1: Overview of benchmark functions

Problems	Functions	Design range	Method	Role
ZDT4	minimize $F(x)=(f_1(x_1),f_2(x))$ $f_1=x_1,\quad f_2=g(x)\left[1-\sqrt{x_1/g(x)}\right]$	\	\	\
ZDT4-R	$g(x)=1+10(D-1)+\sum_{i=1}^{D-1}z_i^2-10\cos(4\pi z_i)$	$x_1\in[0,1]$ $x_i\in[-5,5],i=2,\cdots,D$	NSGA-II	***Model 1*** ***(student)***
ZDT4-G	$g(x)=2+\sum_{i=1}^{D-1}z_i^2/4000-\prod_{i=1}^{D-1}\cos(z_i/\sqrt{i})$	$x_1\in[0,1]$ $x_i\in[-512,512],i=2,\cdots,D$	NSGA-II	***Model 2*** ***(teacher)***
ZDT4-A	$g(x)=21+\exp(1)-20\exp(-0.2\sqrt{\frac{\sum_{i=1}^{D-1}z_i^2}{D-1}})$ $-\exp(\frac{\sum_{i=1}^{D-1}\cos(2\pi z_i)}{D-1})$	$x_1\in[0,1]$ $x_i\in[-32,32],i=2,\cdots,D$	NSGA-II	***Model 2*** ***(teacher)***
ZDT4-RC	constraints : $\cos(\theta)(f_2-e)-\sin(\theta)f_1\geq a\lvert\sin(b\pi(\sin(\theta)(f_2-e)+\cos(\theta)f_1)^c)\rvert^d$ $\theta=-0.05\pi,a=40,b=5,c=1,d=6,e=0$	$x_1\in[0,1]$ $x_i\in[-5,5],i=2,\cdots,D$	NSGA-II	***Model 1*** ***(student)***
(ZDT4-R, ZDT4-G)	Combination of ZDT4-R and ZDT4-G	Being consistent with ZDT4-R and ZDT4-G	PPPFO	\
(ZDT4-A, ZDT4-RC)	Combination of ZDT4-A and ZDT4-RC	Being consistent with ZDT4-A and ZDT4-RC	PPPFO	\

Note: In the variants, $z=M\cdot(x_2,x_3,\cdots,x_D)^T$, where, M is a (D-1)-dimensional square matrix generated randomly.

of combination based on ZDT4-A and ZDT4-RC. The four problems, ZDT4-R/G/A/RC, will be evaluated to determine who play the role of Model 2 or Model 1 based on NSGA-II in the following content. Then, based on the roles of different problems, we can research the performance in the 2-factorial problems based on PPPFO.

4.2. Experimental results and discussion

In order to effectively demonstrate the improvement of optimization performance, fully consistent parameter settings and genetic operations will be performed between PPPFO and NSGA-II. The number of the population in the two paradigms is 100 individuals, which will be evolved for 250 generations. Since the SBX crossover and polynomial mutation are employed in the experiment, the crossover probability (p_c), simulated binary crossover index (Muc), mutation probability (p_m) and polynomial mutation index (Mum) are set to 1 and 10, 1/D and 10, respectively. In the NSGA-II, D=D_i while solving the i^{th} problem. However, in the PPPFO, D= max $\{D_i\}$. In the benchmark functions, the D is set to 10. With regard to the random mating probability, we have set it by online learing (Bali et al., 2020) in the Algorithm 7 to encourage an unobstructed exchange of transfer experience from the past problem to the new problem. The values of the IGD metric reported hereafter are average based on 30 independent runs of the NSGA-II and PPPFO. The parameter settings of the relevant methods have been listed in Table. 2. All experiments were performed on an open-source multitasking optimization platform (MTO-Platform) (Li, 2022).

For the 2-factorial problems of (ZDT4-R, ZDT4-G), we must first determine the role of each problem in the combination. The experimental results are shown in Fig. 10. Note that the normalized IGD curves of ZDT4-R (NSGA-II) and ZDT4-G (NSGA-II) are got based on NSGA-II which only solve a simple problem at a time. It can be seen obviously that ZDT4-G has better convergence performance than ZDT4-R. Accordingly, the ZDT4-G is selected as the Model 2 (past problem like

a teacher) in the 2-factorial problems of (ZDT4-R, ZDT4-G) following the ZDT4-R selected as Model 1 (new problem like a student). The results have been listed in Table. 1.

Table 2: Experimental conditions settings

Experimental conditions	PPPFO	NSGA-II	SPEA2	MOPSO	MOEAD
Parameters setting	Population size:100 Number of iterations:250 Muc:10 Mum:10 ProbSwap:0.5	Population size:100 Number of iterations:250 Muc:10 Mum:10	Population size:100 Number of iterations:250 Muc:10 Mum:10	Population size:100 Number of iterations:250 div:10	Population size:100 Number of iterationss:250 Muc:10 Mum:10 Decomposition Type:1
Hardware environment	11th Gen Intel(R) Core (TM) i7-11700 CPU @ 2.50GHz 2.50 GHz; RAM 16.0GB; SSD 512GB				
Software environment	Win10 Professional Edition; MATLAB R2021a; Win 64 OS; MTO-Platform (Li, 2022)				

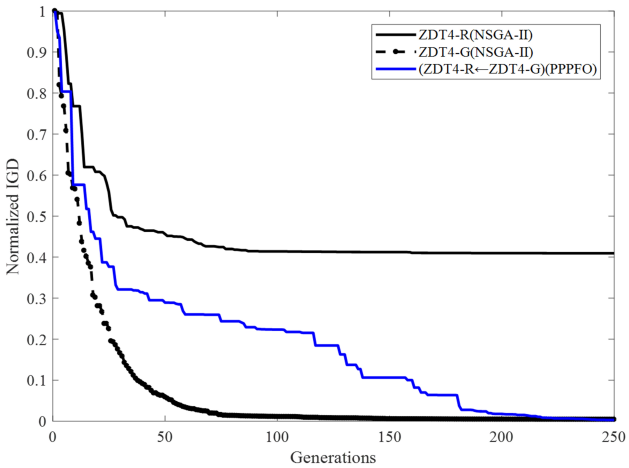

Fig. 10. The averaged convergence characteristics based on normalized IGD metric in the combination of (ZDT4-R, ZDT4-G). (ZDT4-R←ZDT4-G) indicates transferring experience from the past problem (ZDT4-G) to the new problem (ZDT4-R) based on PPPFO.

In fact, according to the function characteristics between ZDT4-R and ZDT4-G, the basin of attraction consistent with every local PF from ZDT4-G is narrow, so this means that searching population can easily break away from the local optimal. However, it a relatively difficult for the searching population to jump out of local optimization because of the wide basins of attraction in ZDT4-R. Then, when we integrate the ZDT4-R and ZDT4-G, we hope ZDT4-G can help speed up the search to improve the convergence performance of ZDT4-R by transferring refined experience (genetic material). In other words, the ZDT4-R can piggyback the past problem (ZDT4-G) for performance improvement. So, the process can be described as (ZDT4-R←ZDT4-G). Fortunately, the experimental result is not disappointing as shown in Fig. 10. The curve of (ZDT4-R←ZDT4-G) based on

PPPFO indicates that convergence characteristics of ZDT4-R in 2-factorial problems has significant improvement than the curve of ZDT4-R(NSGA-II). Another thing worth mentioning is that although both paradigms PPPFO and NSGA-II all use 100 individuals in a population, in essence, PPPFO only uses 50 individuals in a population, because the 100 individuals in PPPFO are divided equally between two problems. In short, the new problem in PPPFO not only uses fewer populations, but also achieves better convergence performance.

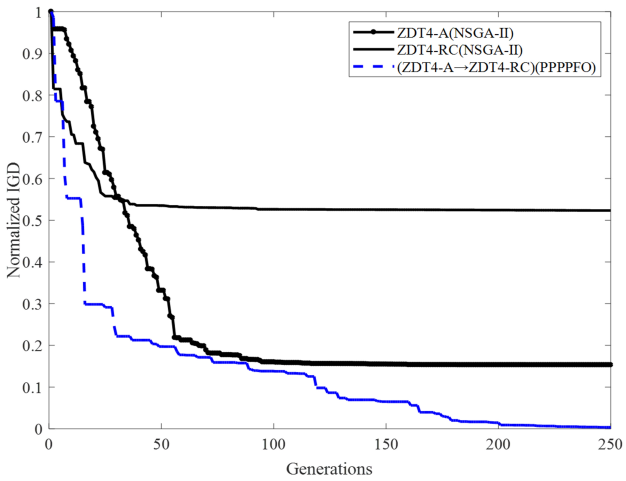

Fig. 11. The averaged convergence characteristics based on normalized IGD metric in the combination of (ZDT4-A, ZDT4-RC). (ZDT4-A→ZDT4-RC) indicates transferring experience from past problem (ZDT4-A) to new problem (ZDT4-RC) based on PPPFO.

For the 2-factorial problems of (ZDT4-A, ZDT4-RC), the numerical results have been depicted in Fig. 11. The ZDT4-A archives a better convergence characteristic than ZDT4-RC. So, in the combination of (ZDT4-A, ZDT4-RC), ZDT4-A and ZDT4-RC will play the role of Model 2 and Model 1, respectively. After implementing the (ZDT4-A→ZDT4-RC) based on PPPFO, the curve of (ZDT4-A→ZDT4-RC) still achieves satisfactory convergence performance than ZDT4-RC (NSGA-II) which only uses NSGA-II.

To further verify whether PPPFO is significantly better than other baselines when optimizing different benchmarks, we let each approach run 30 times independently, and record the performance in terms of average IGD and its standard deviation. We also perform Wilcoxon signed rank test (W-test) (Rosenblatt and Benjamini, 2018) on the four algorithms using the PPPFO as the reference objective to investigate if PPPFO can evidently do better than other baselines. The corresponding results have been summarized in Table. 3. As the table shows, the p-value of PPPFO is far less than 0.05, showcasing that PPPFO is significantly advanced than the compared baselines when optimizing 2-factorial problems, such as (ZDT4-R←ZDT4-G) and (ZDT4-A→ ZDT4-RC).

Meanwhile, a well-known test suit (CEC21-MTMO-CPLX) from the CEC2021 Competition on Evolutionary Multi-task Optimization (CEC2021, 2021) is employed to verify the validity of the proposed framework for complex problems. The test suite for multi-task multi-objective optimization (MTMOO) includes 10 MTO complex problems, in which each problem consists of two multi-objective continuous optimization tasks. The experimental results have been presented in Tables. 4 and 5. It can be seen from the tables, we have labeled the roles of each task (as a teacher or student) based on the framework of PPPFO. According to the experimental results, it indicates that the performance of PPPFO is similar to the three problems (i.e., CPLX1-T1→ CPLX1-T2, CPLX3-T2→ CPLX3-T1 and CPLX10-T1→ CPLX10-T2) compared with other four algorithms and is better than

Table 3: Comparisons in term of IGD among PPPFO (two problems) and other four algorithms (single problem)

Task(s)		PPPFO	NSGA-II	SPEA2	MOPSO	MOEAD
ZDT4-R	Mean	\	0.4094	0.5338	0.2200	0.6069
	Std	\	0.1091	0.1432	0.2555	0.1594
ZDT4-G	Mean	\	0.0039	0.5254	0.6622	0.7424
	Std	\	0.1660	0.1794	0.1560	0.1461
ZDT4-R←ZDT4-G	Mean	**0.0029**	\	\	\	\
	Std	**0.1275**	\	\	\	\
	W-test	**p-value**	**7.71E-04**	**1.25E-04**	**3.50E-03**	**1.13E-05**
ZDT4-A	Mean	\	0.1528	0.3931	0.6513	0.4817
	Std	\	0.2201	0.3359	0.1965	0.223
ZDT4-RC	Mean	\	0.5233	0.6369	0.3117	0.4886
	Std	\	0.0601	0.1808	0.2614	0.2139
ZDT4-A→ZDT4-RC	Mean	**0.0034**	\	\	\	\
	Std	**0.0240**	\	\	\	\
	W-test	**p-value**	**8.47E-06**	**2.12E-06**	**4.70E-03**	**4.33E-06**

them in the remaining 7 complex problems which have been marked in bold font. In addition, similar conclusions can be obtained based on the Rank Sum Test (RST), where the '-' symbol indicates that PPPFO outperforms other algorithms which have been highlighted with a gray background and the '=' indicates that the performance is approximate to other algorithms. The uniqueness of this study is that the framework can effectively identify the roles played by each problem in multitasking problems. Based on the identified roles, it was found after extensive testing experiments that the framework does enable piggybacking on past problems to quickly optimize new problems. The above findings are not explored by existing studies.

How to choose the way of knowledge transfer is an important link in this framework. Therefore, we have specifically investigated the effect of three transfer strategies on the performance of the framework based on the above 10 complex problems. The three strategies are linear transfer, manually setting Random Mating Probability (RMP) and online learning RMP based on MO-MFEA. The experimental performance comparison is shown in Table. 6. It can be seen that the linear transfer way obtains the worst convergence performance and does not outperform the other strategies in any of the 10 problems. For the way of artificially setting RMP, the effect of different RMP on the improvement of optimization performance is shown in detail in the interval [0,1] with 0.2 step length. The experimental results demonstrate that the manually setting RMP is significantly better than the linear transfer, indicating that the convergence performance of the optimization problems can be further improved by an appropriate knowledge transfer strategy. However, there is no significant difference in the optimization performance regardless of what value of RMP is designed in the interval [0,1], indicating that using a fixed RMP is not suitable to participate in optimization experiments. It is obvious that the optimization performance of different problems is effectively improved by using online learning RMP, which has been highlighted in bold. The above experiments demonstrate that the online learning RMP is more appropriate in this framework to accomplish the knowledge transfer from the source task to the target task.

Based on the above analysis, the benchmark function (new problem like a student) can significantly improve the convergence characteristics by piggybacking the past problem in the evolutionary 2-factorial environment. In other words, the implicit genetic transfer can bring a positive effect to accelerate convergence achieved by PPPFO. So, this is an interesting proposition by piggybacking

Table 4: Comparisons in term of IGD among PPPFO (two problems) and other four algorithms (single problem) based on CEC21-MTMO-CPLX(1-5) benchmarks

Tasks	Indicator	PPPFO	NSGA-II	SPEA2	MOPSO	MOEAD
CPLX1-T1(Teacher)	Mean	\	5.01E+06	3.87E+06	2.66E+06	4.15E+06
	Std	\	3.09E+06	2.03E+06	1.26E+06	2.38E+06
CPLX1-T2(Student)	Mean	\	5.53E+06	3.76E+06	8.22E+06	3.37E+06
	Std	\	3.54E+06	2.44E+06	6.98E+06	1.84E+06
CPLX1-T1→ CPLX1-T2	Mean	3.92E+06	\	\	\	\
	Std	2.10E+06	\	\	\	\
	RST	Base	=	=	=	=
CPLX2-T1(Student)	Mean	\	5.79E+01	4.75E+01	1.12E+02	3.99E+01
	Std	\	2.81E+01	2.29E+01	3.47E+01	1.82E+01
CPLX2-T2(Teacher)	Mean	\	4.73E+01	6.61E+01	1.14E+02	4.88E+01
	Std	\	2.11E+01	3.64E+01	2.32E+01	2.85E+01
CPLX2-T2→ CPLX2-T1	Mean	**3.52E+01**	\	\	\	\
	Std	**2.55E+00**	\	\	\	\
	RST	Base	-	-	-	=
CPLX3-T1(Student)	Mean	\	1.41E+03	1.25E+03	1.04E+03	1.90E+03
	Std	\	2.68E+02	2.92E+02	3.02E+02	3.98E+02
CPLX3-T2(Teacher)	Mean	\	1.23E+03	1.21E+03	1.59E+03	1.45E+03
	Std	\	3.39E+02	2.57E+02	4.28E+02	4.12E+02
CPLX3-T2→ CPLX3-T1	Mean	1.33E+03	\	\	\	\
	Std	4.72E+02	\	\	\	\
	RST	Base	=	=	=	-
CPLX4-T1(Student)	Mean	\	8.79E+01	8.08E+01	2.45E+04	6.01E+01
	Std	\	5.04E+01	2.96E+01	1.34E+04	1.65E+01
CPLX4-T2(Teacher)	Mean	\	1.61E+02	1.81E+02	4.81E+04	6.96E+01
	Std	\	1.05E+02	1.34E+02	4.64E+04	1.27E+01
CPLX4-T2→ CPLX4-T1	Mean	**2.32E+01**	\	\	\	\
	Std	**5.23E+00**	\	\	\	\
	RST	Base	-	-	-	-
CPLX5-T1(Student)	Mean	\	4.39E+02	4.28E+02	3.04E+03	2.22E+02
	Std	\	1.10E+02	1.03E+02	6.56E+02	6.67E+01
CPLX5-T2(Teacher)	Mean	\	3.59E+02	4.46E+02	3.81E+03	1.36E+02
	Std	\	1.08E+02	1.17E+02	1.19E+03	6.84E+01
CPLX5-T2→ CPLX5-T1	Mean	**1.65E+02**	\	\	\	\
	Std	**5.40E+01**	\	\	\	\
	RST	Base	-	-	-	-

Table 5: Comparisons in term of IGD among PPPFO (two problems) and other four algorithms (single problem) based on CEC21-MTMO-CPLX(6-10) benchmarks

CPLX6-T1(Student)	Mean	\	3.79E+02	3.43E+02	2.70E+02	4.40E+02
	Std	\	4.13E+01	4.47E+01	3.08E+01	4.56E+01
CPLX6-T2(Teacher)	Mean	\	3.77E+02	3.54E+02	2.87E+02	4.86E+02
	Std	\	2.53E+01	3.76E+01	2.61E+01	4.82E+01
CPLX6-T2→ CPLX6-T1	Mean	**1.11E+02**	\	\	\	\
	Std	**2.90E+01**	\	\	\	\
	RST	Base	-	-	-	-
CPLX7-T1(Student)	Mean	\	2.07E+02	2.10E+02	2.65E+02	1.23E+02
	Std	\	2.78E+01	2.96E+01	2.04E+01	4.44E+01
CPLX7-T2(Teacher)	Mean	\	1.56E+02	1.61E+02	3.21E+02	2.40E+01
	Std	\	3.07E+01	3.68E+01	3.70E+01	5.11E+00
CPLX7-T2→ CPLX7-T1	Mean	**1.34E+01**	\	\	\	\
	Std	**3.55E+00**	\	\	\	\
	RST	Base	-	-	-	-
CPLX8-T1(Student)	Mean	\	1.68E+05	4.68E+05	9.54E+08	2.13E+04
	Std	\	1.27E+05	4.06E+05	2.94E+08	7.08E+03
CPLX8-T2(Teacher)	Mean	\	5.27E+04	5.47E+04	1.53E+05	4.51E+04
	Std	\	2.00E+04	2.21E+04	1.95E+05	1.21E+04
CPLX8-T2→ CPLX8-T1	Mean	**4.04E+03**	\	\	\	\
	Std	**4.00E+03**	\	\	\	\
	RST	Base	-	-	-	-
CPLX9-T1(Teacher)	Mean	\	6.76E+03	7.08E+03	1.01E+04	8.11E+03
	Std	\	9.23E+02	7.64E+02	5.95E+02	5.78E+02
CPLX9-T2(Student)	Mean	\	9.26E+04	1.04E+05	4.70E+09	6.86E+03
	Std	\	6.51E+04	6.00E+04	2.04E+09	3.66E+03
CPLX9-T1→CPLX9-T2	Mean	**3.49E+03**	\	\	\	\
	Std	**2.72E+03**	\	\	\	\
	RST	Base	-	-	-	-
CPLX10-T1(Teacher)	Mean	\	1.11E+02	8.65E+01	3.15E+04	6.15E+01
	Std	\	4.80E+01	2.52E+01	1.67E+04	1.02E+01
CPLX10-T2(Student)	Mean	\	6.30E+06	4.22E+06	5.97E+07	3.78E+06
	Std	\	2.92E+06	2.63E+06	4.63E+07	2.46E+06
CPLX10-T1→ CPLX10-T2	Mean	5.19E+06	\	\	\	\
	Std	3.41E+06	\	\	\	\
	RST	Base	=	=	-	=

Table 6: Performance comparison of the important parameter RMP and ablation experiments based on PPPFO

Tasks	Indicator	Linear transfer	RMP						Online learning
			0	0.2	0.4	0.6	0.8	1	
CPLX1-T1→ CPLX1-T2	Mean	5.18E+06	4.25E+06	4.79E+06	3.53E+06	**2.79E+06**	3.99E+06	3.99E+06	3.92E+06
	Std	2.93E+06	2.66E+06	1.99E+06	**1.31E+06**	1.37E+06	1.68E+06	2.41E+06	2.10E+06
CPLX2-T2→ CPLX2-T1	Mean	6.27E+01	5.49E+01	4.99E+01	5.63E+01	5.57E+01	4.56E+01	5.14E+01	**3.50E+01**
	Std	2.42E+01	2.50E+01	2.33E+01	2.57E+01	2.13E+01	1.74E+01	2.30E+01	**2.55E+00**
CPLX3-T2→ CPLX3-T1	Mean	2.36E+04	1.45E+03	1.17E+03	1.22E+03	1.21E+03	1.44E+03	**1.13E+03**	1.33E+03
	Std	3.57E+02	2.95E+02	3.95E+02	4.60E+02	3.71E+02	4.30E+02	**2.58E+02**	4.72E+02
CPLX4-T2→ CPLX4-T1	Mean	2.61E+02	1.07E+02	9.51E+01	8.19E+01	1.10E+02	1.05E+02	1.44E+02	**2.32E+01**
	Std	4.83E+01	6.37E+01	3.82E+01	2.50E+01	5.23E+01	3.66E+01	1.27E+02	**5.23E+00**
CPLX5-T2→ CPLX5-T1	Mean	6.34E+02	4.80E+02	4.62E+02	4.93E+02	4.54E+02	4.80E+02	4.63E+02	**1.65E+02**
	Std	5.81E+01	7.56E+01	1.08E+02	8.85E+01	8.58E+01	6.74E+01	1.37E+02	**5.36E+01**
CPLX6-T2→ CPLX6-T1	Mean	4.93E+02	3.78E+02	3.64E+02	3.67E+02	3.90E+02	3.64E+02	4.01E+02	**1.11E+02**
	Std	3.57E+01	**2.23E+01**	2.93E+01	5.60E+01	2.69E+01	3.94E+01	4.89E+01	2.90E+01
CPLX7-T2→ CPLX7-T1	Mean	3.51E+02	1.88E+02	2.15E+02	2.19E+02	2.22E+02	2.59E+02	2.86E+02	**1.34E+01**
	Std	3.45E+01	2.72E+01	3.08E+01	3.03E+01	2.47E+01	3.72E+01	2.94E+01	**3.55E+00**
CPLX8-T2→ CPLX8-T1	Mean	5.72E+07	2.69E+05	3.81E+05	4.89E+05	6.50E+05	1.27E+06	2.52E+06	**4.04E+03**
	Std	2.31E+06	1.63E+05	2.88E+05	2.02E+05	4.89E+05	7.59E+05	1.32E+06	**4.00E+03**
CPLX9-T1→ CPLX9-T2	Mean	3.16E+05	2.24E+05	1.56E+05	2.32E+05	3.18E+05	5.15E+05	5.84E+05	**3.49E+03**
	Std	5.74E+05	1.64E+05	9.49E+04	1.83E+05	1.90E+05	4.82E+05	2.72E+05	**2.72E+03**
CPLX10-T1→ CPLX10-T2	Mean	7.98E+07	6.01E+06	6.63E+06	5.78E+06	6.95E+06	**4.80E+06**	7.02E+06	5.19E+06
	Std	4.73E+07	**2.63E+06**	4.97E+06	3.95E+06	3.86E+06	3.48E+06	3.53E+06	3.41E+06

Table 7: Selection of design variables and target variables

Types of variables	Variables	Description
Design variables	x_1	series current (A)
	x_2	molecular ratio (1)
	x_3	aluminum level (cm)
	x_4	electrolyte level (cm)
	x_5	cell temperature (^{o}C)
	x_6	amount of aluminum extracted (kg)
	x_7	daily dosage of salt fluoride (kg)
	x_8	NB times (s)
	x_9	working voltage (v)
Target variables	f_1	current efficiency (1)
	f_2	direct current consumption of aluminum per ton (kW·h /t-Al)

past problem for faster optimization to the new problem based on the framework of PPPFO and will give us more inspiration to solve practical industrial problems.

5. Real application of PPPFO in aluminum electrolysis process design

To investigate whether the proposed PPPFO is able to solve the complex optimization problems in real-world, we used PPPFO to optimize the design of new devices in aluminum electrolysis and analyzed the performance of PPPFO. Using the mentioned industrial process as the testing case is mainly because device updating is a typical behavior in aluminum electrolysis process and the proposal of PPPFO is mainly motivated by the fact that the newly updated devices in aluminum electrolysis are almost same with the old, except for several key components (depicted in Fig. 1).

5.1. Variable selection of aluminum electrolytic process

The aluminum electrolytic process involves complex physical and chemical changes with the characteristics such as many variables and strong coupling among them. It is a very difficult thing to find out the mechanism of this process. So, these variables are selected based on expert experience. The variable selection includes selecting design variables and target variables. The selection results are as shown in Table. 7.

5.2. Problem identification

The old device is the mainstream equipment of aluminum electrolysis and has accumulated a wealth of valuable design experience. However, the new device is currently under the experimental production stage and lack excellent design experience. So, the new device is viewed as Model 1 (new problem like a student) and the old device is treated as Model 2 (past problem like a teacher). Compared the new device with the old, the target variables both are f_1 and f_2. Moreover, the design parameters are identical and complete overlap, Xn =Xp. However, the design ranges between DRnew and DRpast are different because of upgrading and improvement of the old device as listed in Table. 8.

5.3. Modeling for past and new problems

In this section, BP neural network is used to establish the models of the past problem and new problem. Detailed algorithm can be found in Algorithm 2. In the modeling, all the sample data

Table 8: Design ranges of past and new problems

Problems	Bound	Design variables								
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Past problem	Lower Bound	1655	2.26	21	15	928	1210	0	545	3784
	Upper Bound	1698	2.71	28	28	968	1360	382	1264	3852
New problem	Lower Bound	1636	2.26	19.5	12	931	1000	0	540	3766
	Upper Bound	1698	2.73	25.5	20	977	1350	334	742	3848

Table 9: The basic modeling parameters

Types	Past problem (Model 2)	New problem (Model 1)
Epochs	10000	10000
Learning rate	0.1	0.1
Minimum error of training goal	10^{-5}	10^{-5}
Transfer function of hidden layer	Logsig	Logsig
Transfer function of output layer	Purlin	Purlin
Number of hidden layer neurons	***7***	***9***
Inputs	$x_i, i = 1, 2, \cdots, 9$	$x_i, i = 1, 2, \cdots, 9$
Outputs	f_1 and f_2	f_1 and f_2

from the old device and new device are captured from the real-time database of Chongqing Tian Tai Aluminum Industry Corporation Ltd, located in the southwest of China. The sample set of every problem contains 332 samples, which is grouped into training set with 282 samples and test set with 50 samples. The basic modeling parameters are listed in Table. 9. It should be noted that, in our experiments, the standard logistic sigmoid function is used in the hidden layers of the neural network (NN). Adopting such a generic setting is due to the following reasons. First, logistic sigmoid function is one of the most classical activation functions for NNs. It has several important properties, e.g., monotonously increasing, continuous, differentiable, and output ranging between 0 and 1, which are desirable for modeling the real problems presented in the paper. Second, as in this research, we mainly investigate whether the newly-designed mechanism for knowledge transfer may effectively improve the optimization performance of generic framework used in aluminum electrolysis, we did not particularly re-design the NN structure, which might lead the improvement of optimization to be hard to evaluate. With regard to the number of hidden layer neurons, it is selected based on Algorithm 3. According to equation (5) and Table. 9, n should be set from 5 to 14. So, the MSE from different numbers of hidden layer neurons is shown in Fig. 12 and Fig. 13. As can be seen from Fig. 12, when the number of hidden layer neurons is set to 7, the past problem (Model 2) gets the better prediction performance. Meanwhile, Fig. 13 shows that 9 hidden layer neurons should be selected for the new problem. The selection results have been listed in the Table. 9.

Based on Algorithm 4 to exploit best models of the past and new problems, the Fig. 14 demonstrates the prediction errors of target variables f_1 and f_2 from the two problems respectively. The prediction errors of f_1 and f_2 in the past problem are shown in Fig. 14(a) and (b), which present approximately the maximum prediction errors of them, 1.3% and 1.9%, respectively. In Fig. 14(c) and (d), the maximum prediction errors of f_1 and f_2 from the new problem are about -1.67% and 0.93%, respectively. For aluminum electrolysis process design, the predictive performance of the

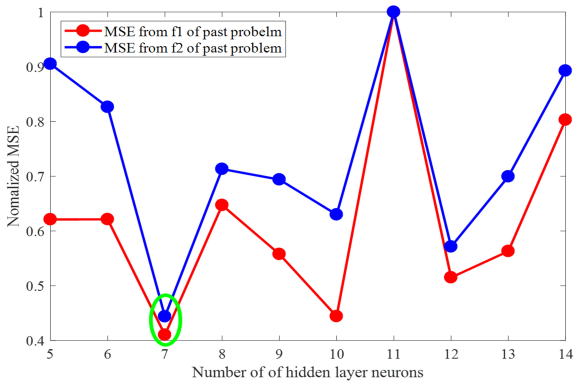

Fig. 12. The MSE from past problem based on different numbers of hidden layer neurons

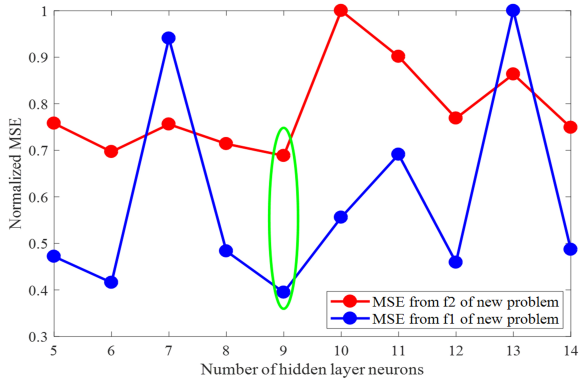

Fig. 13. The MSE from new problem based on different numbers of hidden layer neurons

above models is sufficient to meet the accuracy requirements.

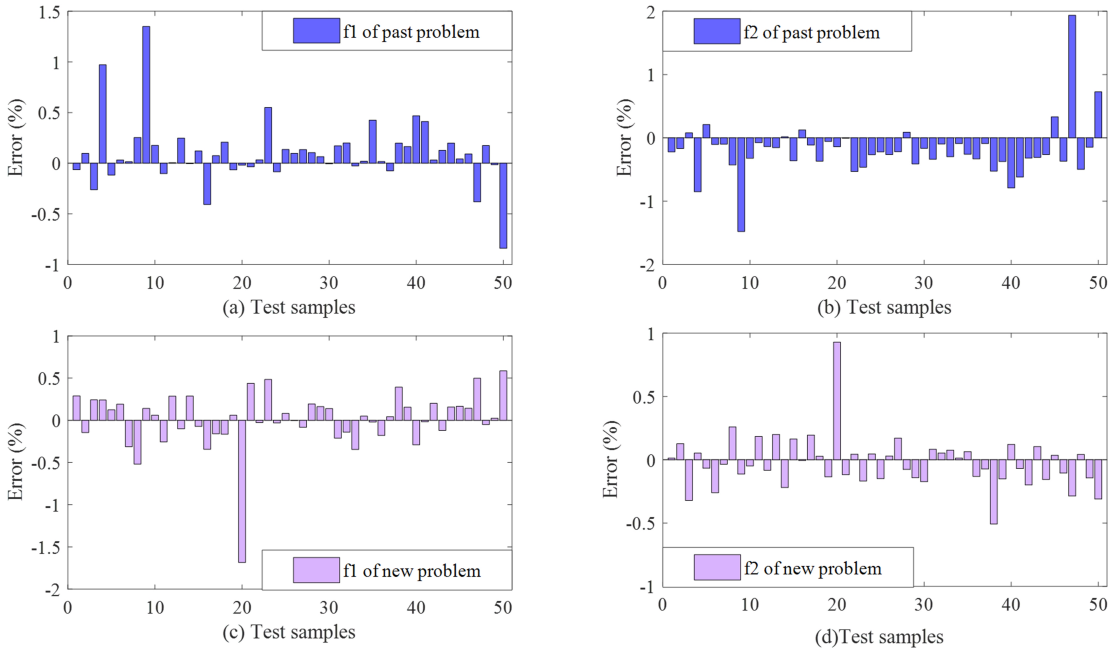

Fig. 14. Prediction errors of past and new problems based on test samples

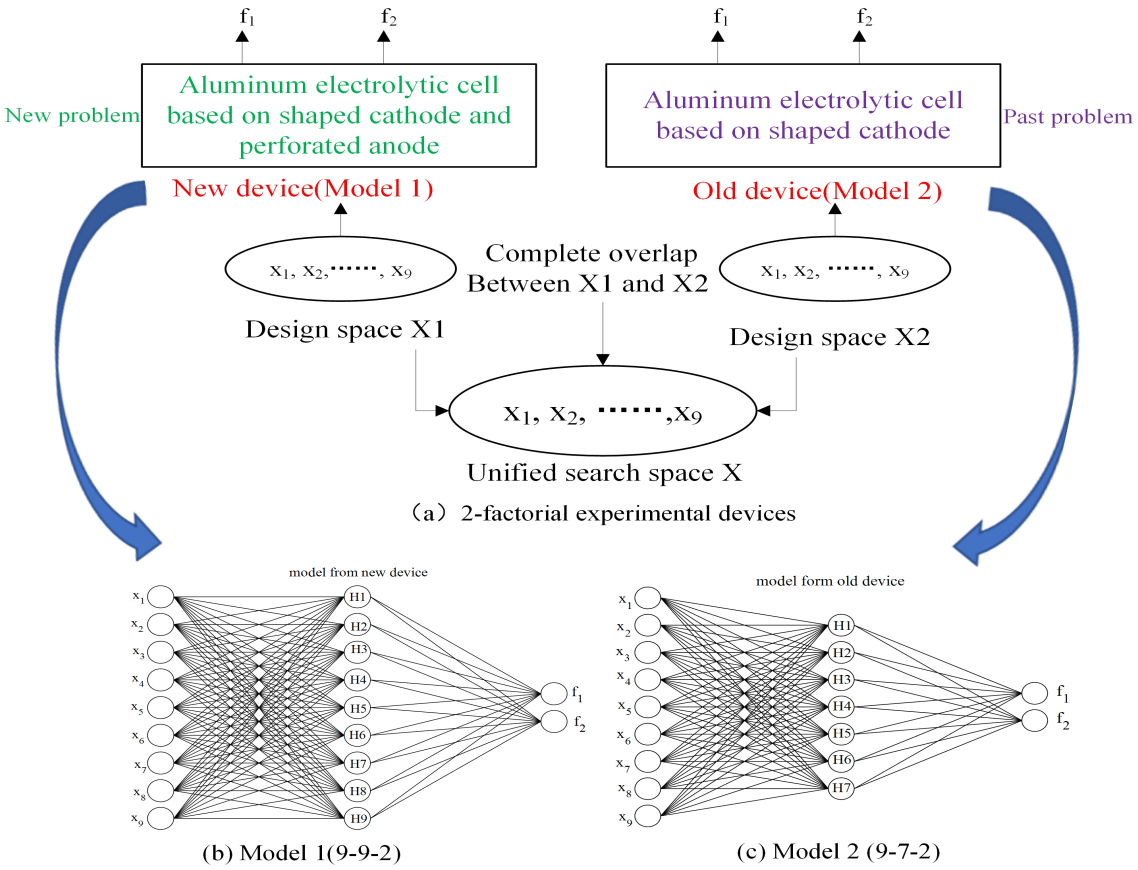

Fig. 15. The core experimental structure in the aluminum electrolysis process design

5.4. Optimization process and results

The ultimate expectation in this study is piggybacking past problem for faster optimization of new problem. So, after completing all the above works, the core experimental structure in the aluminum electrolysis process design is shown clearly in Fig. 15. Fig. 15(a) visually presents 2-factorial experimental devices fully matching this paradigm of PPPFO, in which one represents a past problem, the other represents a new problem. In the paradigm of PPPFO, the new problem will get transfer experience from the past problem. With regard to design spaces, when they are completed overlap or partial overlap, the two situations are very popular with this paradigm. In our case study, the design space is completely overlapping. For the models, (9-9-2) in Fig. 15(b) depicts the structure of the Model 1. (9-7-2) in Fig. 15(c) depicts the structure of the Model 2. Both models have 2 target variables f_1 and f_2 .

In the aluminum electrolysis process design, an ideal result is to maximize current efficiency f_1 and minimize process energy consumption f_2. Without loss of generality, for a general multi-objective minimization problem, a better practice is to set f_1 to r/f_1 , where r is a lager constant to avoid calculation loss when the $1/f_1$ is too small. To this end, the optimization problem of each device can be described as,

$$\begin{aligned} &minimize \quad (r/f_1, f_2) \\ &subjectto: \quad f_1 \leq 100 \end{aligned} \tag{6}$$

Where, r/f_1 can't be greater than 100, because it is in theory for a real application.

Table 10: Injection ways and operations

Injection way	Specific operation
Injection 1	Injections of past optimal designs from every generation of the past problem into corresponding generation of the new device
Injection 2	Injections of past optimal designs from the last generation of past problem into the first generation (as the initial value) of the new device
Injection 3	Injections of 10 past optimal designs from the last generation of past problem into the first generation (as the initial value) of the new device
Injection 4	Injections of 10 past optimal designs from every generation of past problem into the corresponding generation (replacing the worst 10 individuals) of the new device

5.4.1. Configuration of optimization parameters

The number of the population is 50 individuals both in NSGA-II and PPPFO, which will be evolved for 100 generations. Parameters of r, p_c , η_c , p_m, η_m are set to 1000, 1, 10, 1/9 and 10, respectively. The random mating probability is set based on the Algorithm 7 to encourage the unobstructed exchange of transfer experience from Model 2 to Model 1. Since the real pareto front is unknown, Hypervolume (HV) metric is employed in the case. The reference point is set to (50,16000). The values of the HV metric reported hereafter are averaged based on 30 independent runs of the NSGA-II and PPPFO.

5.4.2. Optimized results

To effectively compare the optimization performance of different paradigms, we first implement a set of experiments based on linear transfer as described in Algorithm 5 from the past problem to the new one. It is a fairer way to complete the transfer experience in the genotype space described with random-keys (from key_i^p to key_i^n) in Section 3. The experimental results are as follows.

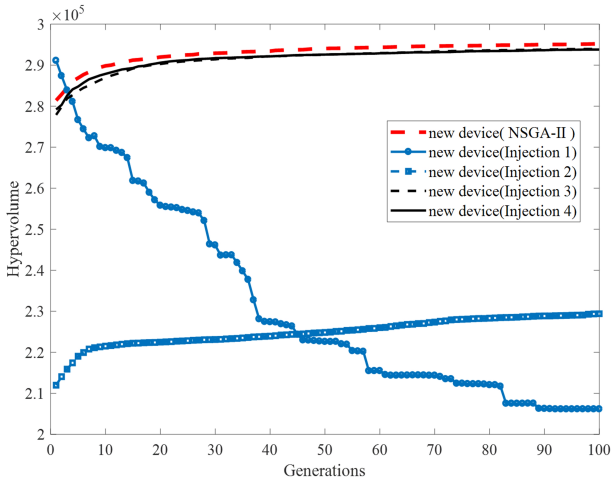

Fig. 16. Comparing the evolution of the HV metric for the case of new device. Note that the injection n (n=1, 2, 3, 4) denotes a particular task by using injections of past optimal designs of the past problem into the new problem in different ways, as shown in Table. 10.

In Fig. 16, the red line from the new device shows the evolved performance of HV metric based on NSGA-II. To speed up the optimization process, some ways of transfer experience are executed to explore the possibility of further improving performance. The Injections 1 and 2 in Fig. 16 show that it is completely unworkable to inject directly all the past optimal solutions to the new problem.

The performances of HV metric using the two ways are far from the red line based on NSGA-II. It indicates that the new device doesn't receive effective transfer experience from the past problem. Meanwhile, it can be seen from Injections 3 and 4 that the performances of HV metric based on the two ways have significant improvement than Injections 1 and 4. Although all the results of Injection n (n= 1, 2, 3, 4) are with worse performance than the new device (NSGA-II), it also shows that Injections 3 and 4 have better performance than Injections 1 and 2. This phenomenon indicates that the convergence characteristic of new device could be improved by using injections of past optimal designs of old problem into the new device in a proper way. The nature of the phenomenon may tell us that it has great potential to further explore the performance of a problem and help speed up its search (in the new device) if we can make full use of the past design experience (as is the case in the old device).

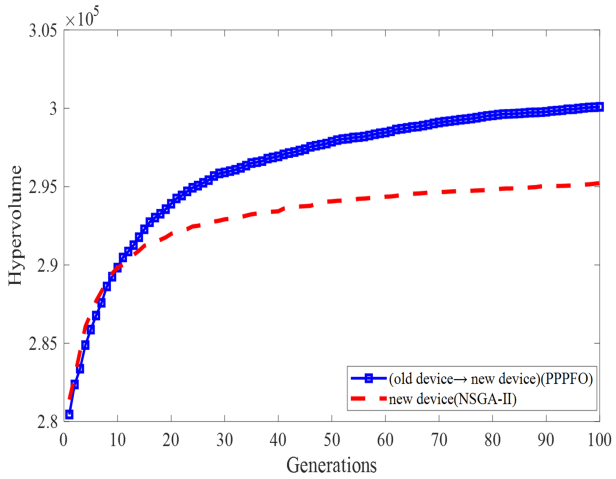

Fig. 17. Comparing the evolution of the HV metric for the case of new device. Note that the "old device → new device" indicates transferring experience from past problem (old device) to new problem (new device) based on PPPFO

Fortunately, the PPPFO by introducing skill factor and random mating probability in genetic operator (see Algorithm 7) is helpful to achieve implicit genetic transfer from the past problem for excavating hidden potential in the new device. (old device→ new device) (PPPFO) in Fig. 17 shows that the overall convergence characteristics of new device in the PPPFO can be notably boosted as opposed to challenging a single optimization problem at a time with NSGA-II. (In Fig. 17, the red line from the new device (NSGA-II) is identical to the red line in Fig. 16). Similar to what has been done in Section 4.2, here we also investigate whether PPPFO is significantly better than other baselines. The corresponding results are presented in Table. 11. As the table shows, the performance of PPPFO is significantly better than that obtained by other baselines. Based on the above analyses, the effectiveness of the proposition by piggybacking past problem for faster optimization has been proved once again. For real industrial applications (i.e., aluminum electrolysis process design), the proposed framework not only improves production efficiency, but also reduces production energy consumption. More importantly, it can bring more benefits to enterprises and society.

Strictly speaking, PPPFO is a framework used to piggyback on the past optimization problem for a faster convergence of the targeted, in which MO-MFEA is a component. The main advantage of PPPFO is that for specific engineering optimization design problems, a complete implementation process and operation plan are given, including the design variables selection, optimization model construction and roles identification, knowledge transfer and performance evaluation, etc.,

Table 11: Comparisons in term of HV between (old device → new device) and new device (single problem)

Task(s)		PPPFO	NSGA-II	SPEA2	MOPSO	MOEAD
old device	Mean	\	2.92E+05	2.83E+05	2.76E+05	2.79E+05
	Std	\	3.19E+03	3.71E+03	4.72E+03	8.46E+03
new device	Mean	\	2.95E+05	2.86E+05	2.86E+05	2.82E+05
	Std	\	5.33E+03	5.59E+03	4.55E+03	4.80E+03
old device→new device	Mean	**3.00E+5**	\	\	\	\
	Std	**3.59E+3**	\	\	\	\
	W-test	**p-value**	**2.80E-03**	**2.80E-03**	**1.50E-03**	**4.73E-06**

where the knowledge transfer is built on the basis of roles identification focusing on the performance optimization of new problems. These are the advantageous features of PPPFO framework. If those MO-MFEA algorithms with outstanding optimization performance can be embedded into the PPPFO framework, it is expected to further enhance the applicability and application scope of the framework. But, the specific form of its adaptation to PPPFO needs to be explored in depth.

5.5. Limitations of the study and future directions

As reported by benchmark problems and the real industrial application, the framework presented in this paper is significantly different from existing patterns for the optimal design of aluminum electrolysis process parameters. This study proposes a new perspective to piggyback past problems (old devices) to rapidly optimize new problems (new devices). Although the present framework achieves better optimization performance compared to the conventional optimization methods, there are still the following limitations that need to be further overcome. (1) The optimization performance of this framework relies heavily on the parameter settings, which need to be constantly updated to achieve the desired goal. However, this may lead to a deepening of the algorithm complexity. Therefore, designing a multitasking algorithm with outstanding optimization ability and low complexity is an area for future exploration. (2) This study focuses on solving a more general class of problems with 2-task optimal design in industrial practice, where one task is an old problem and the other is new. The exploration of more tasks with different roles and objectives is lacking. Subsequently, the effectiveness of the proposed framework can be further explored in this area. (3) The paper (Akyol and Alatas, 2017) discussed 9 different metaheuristic optimization algorithms, e.g., biology-based, physic-based, swarm-based, social-based, music-based, chemistry-based, sport-based, math-based and hybrid methods. According to the above classification, the PPPFO framework presented in the paper is biology based. So far, existing multitasking optimization frameworks are typically biology-based and swarm-based. Multitasking optimization frameworks based on other metaheuristic types (i.e., physics based and sport based) are less studied. Inspired by the paper (Akyol and Alatas, 2017), exploring other metaheuristic algorithms in combination with multitask optimization frameworks would be a very cutting-edge research perspective.

6. Conclusions

A phenomenon is often observed in real engineering design, that is the newly upgraded devices or processes in the initial stage fail to take advantage of rich design experiences accumulated by the previously similar industrial installations or processes. As a result, finding the optimal settings of the upgraded devices always becomes a time-demanding task. To address this problem, a novel framework for accelerating the convergence of new optimization problems by means of deriving beneficial

experiences from those problems possessing similar optimization properties, dubbed piggybacking past problem for faster optimization (PPPFO), is proposed in this paper. For aluminum electrolysis process parameter design, PPPFO is first proposed to pickaback the old devices to quickly optimize new newer devices. This study is a bold attempt to construct a complete multi-task optimization framework to compare the optimization effect with the traditional multi-objective optimization methods which are only used to optimize a device. The proposed PPPFO is extensively tested with a number of widely used benchmarking functions and has been applied in aluminum electrolysis to efficiently seek for the optimal settings of the upgraded electrolyzer. The experimental results show that PPPFO outperforms any other baseline in all the experiments, which validates PPPFO's strong capability of guaranteeing a fast convergence of related optimization problems via an effective scheme of knowledge transfer. This paper will help engineering designers and related researchers to quickly build optimized design solutions and find near-optimal design parameters for multi-tasking cases with different roles. In future, we will explore combining other metaheuristic algorithms (i.e., physics based) and multitasking optimization problems with different roles in order to obtain a better optimization mechanism. Meanwhile, we plan to thematically investigate the embedding of relevant multi-task optimization algorithms into the PPPFO framework and comprehensively analyze the impact of the framework on the field of aluminum electrolysis engineering design. Based on this new vision, it is expected that PPPFO can be extended to the setting, which allows optimization experiences to transfer from multiple teachers to many students.

Acknowledgments

The work was supported in part by the Data Science and Artificial Intelligence Research Centre (DSAIR) and the School of Computer Science and Engineering at Nanyang Technological University, Singapore, in part by the National Natural Science Foundation of China (No.51805059), in part by the Foundation Program of Chongqing Normal University (No.22XLB014), and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission (No. KJQN202200531).

References

Akyol, S., Alatas, B., 2017. Plant intelligence based metaheuristic optimization algorithms. Artificial Intelligence Review 47, 417–462.

Andrade, C.E., Toso, R.F., Gonçalves, J.F., Resende, M.G., 2021. The multi-parent biased random-key genetic algorithm with implicit path-relinking and its real-world applications. European Journal of Operational Research 289, 17–30.

Bali, K.K., Gupta, A., Feng, L., Ong, Y.S., Siew, T.P., 2017. Linearized domain adaptation in evolutionary multitasking, in: 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE. pp. 1295–1302.

Bali, K.K., Gupta, A., Ong, Y.S., Tan, P.S., 2021. Cognizant multitasking in multiobjective multifactorial evolution: Mo-mfea-ii. IEEE Transactions on Cybernetics 51, 1784–1796.

Bali, K.K., Ong, Y.S., Gupta, A., Tan, P.S., 2020. Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii. IEEE Transactions on Evolutionary Computation 24, 69–83.

Binh, H.T.T., Thang, T.B., Thai, N.D., Thanh, P.D., 2021. A bi-level encoding scheme for the clustered shortest-path tree problem in multifactorial optimization. Engineering Applications of Artificial Intelligence 104187.

CEC2021, 2021. CEC2021 Competition on Evolutionary Multi-task Optimization http://www.bdsc.site/websites/MTO_competition_2021/MTO_Competition_CEC_2021.html.

Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K., 2016. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Transactions on Evolutionary Computation 22, 129–142.

Cloninger, C.R., Rice, J., Reich, T., 1979a. Multifactorial inheritance with cultural transmission and assortative mating. ii. a general model of combined polygenic and cultural inheritance. American journal of human genetics 31, 176–198.

Cloninger, C.R., Rice, J., Reich, T., 1979b. Multifactorial inheritance with cultural transmission and assortative mating. iii. family structure and the analysis of separation experiments. American Journal of Human Genetics 31, 366–388.

Deb, K., Hussein, R., Roy, P.C., Toscano-Pulido, G., 2019. A taxonomy for metamodeling frameworks for evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation 23, 104–116.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation 6, 182–197.

Ding, J., Yang, C., Jin, Y., Chai, T., 2019. Generalized multitasking for evolutionary optimization of expensive problems. IEEE Transactions on Evolutionary Computation 23, 44–58.

Ding, W., Yao, L., Li, Y., Long, W., Yi, J., He, T., 2021. Dynamic evolutionary model based on a multi-sampling inherited hapfnn for an aluminium electrolysis manufacturing system. Applied Soft Computing 99, 106925.

Feng, L., Huang, Y., Zhou, L., Zhong, J., Gupta, A., Tang, K., Tan, K.C., 2021. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. IEEE transactions on cybernetics 51, 3143–3156.

Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y.S., Tan, K.C., Qin, A.K., 2019. Evolutionary multitasking via explicit autoencoding. IEEE transactions on cybernetics 49, 3457–3470.

Gao, W., Cheng, J., Gong, M., Li, H., Xie, J., 2022. Multiobjective multitasking optimization with subspace distribution alignment and decision variable transfer. IEEE Transactions on Emerging Topics in Computational Intelligence 6, 818–827.

Gonçalves, J.F., Resende, M.G., 2011. Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics 17, 487–525.

Gong, M., Li, H., Meng, D., Miao, Q., Liu, J., 2018. Decomposition-based evolutionary multiobjective optimization to self-paced learning. IEEE Transactions on Evolutionary Computation 23, 288–302.

Gong, M., Tang, Z., Li, H., Zhang, J., 2019. Evolutionary multitasking with dynamic resource allocating strategy. IEEE Transactions on Evolutionary Computation , 858–869.

Gui, W., Yue, W., Xie, Y., Zhang, H., Yang, C., 2018. A review of intelligent optimal manufacturing for aluminum reduction production. Zidonghua Xuebao/Acta Automatica Sinica 44, 1957–1970.

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

Gupta, A., Ong, Y.S., Feng, L., 2016. Multifactorial evolution: toward evolutionary multitasking. IEEE Transactions on Evolutionary Computation 20, 343–357.

Gupta, A., Ong, Y.S., Feng, L., 2018. Insights on transfer optimization: Because experience is the best teacher. IEEE Transactions on Emerging Topics in Computational Intelligence 2, 51–64.

Gupta, A., Ong, Y.S., Feng, L., Tan, K.C., 2017. Multiobjective multifactorial optimization in evolutionary multitasking. IEEE transactions on cybernetics 47, 1652–1665.

Hao, X., Qu, R., Liu, J., 2021. A unified framework of graph-based evolutionary multitasking hyper-heuristic. IEEE Transactions on Evolutionary Computation 25, 35–47.

Haraldsson, J., Johansson, M.T., 2018. Review of measures for improved energy efficiency in production-related processes in the aluminium industry–from electrolysis to recycling. Renewable and Sustainable Energy Reviews 93, 525–548.

Hou, W., Li, H., Feng, Y., Wang, J., Cao, X., 2020. Effects of the application of a perforated anode in an aluminum electrolysis cell on the gascliquid two-phase flow and bubble distribution characteristics. Industrial and Engineering Chemistry Research 59, 14522–14530.

Ji, X., Zhang, Y., Gong, D., Sun, X., Guo, Y., 2023. Multisurrogate-assisted multitasking particle swarm optimization for expensive multimodal problems. IEEE Transactions on Cybernetics 53, 2516–2530.

Jiang, S., Ong, Y.S., Zhang, J., Feng, L., 2014. Consistencies and contradictions of performance metrics in multiobjective optimization. IEEE transactions on cybernetics 44, 2391–2404.

Li, B., Wang, F., Zhang, X., Qi, F., Feng, N., 2012. Modeling of interface of electrolyte/aluminum melt in aluminum reduction cell with novel cathode structure, in: Light Metals 2012. Springer, pp. 865–868.

Li, G., Lin, Q., Gao, W., 2020a. Multifactorial optimization via explicit multipopulation evolutionary framework. Information Sciences 512, 1555–1570.

Li, H., Wang, J., Hou, W., Li, M., Cheng, B., Feng, Y., Xu, T., 2021. The study of carbon recovery from electrolysis aluminum carbon dust by froth flotation. Metals 11, 145.

Li, K., Liao, M., Deb, K., Min, G., Yao, X., 2020b. Does preference always help? a holistic study on preference-based evolutionary multi-objective optimisation using reference points. IEEE Transactions on Evolutionary Computation 24, 1078–1096.

Li, Y., 2022. MTO-Platform https://github.com/intLyc/MTO-Platform.

Liang, Z., Dong, H., Liu, C., Liang, W., Zhu, Z., 2022. Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution. IEEE Transactions on Cybernetics 52, 2096–2109.

Liang, Z., Zhang, J., Feng, L., Zhu, Z., 2019. A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking. Expert Systems with Applications 23, 858C869.

Liu, H., Ong, Y., Cai, J., Wang, Y., 2018. Cope with diverse data structures in multi-fidelity modeling: A gaussian process method. Engineering Applications of Artificial Intelligence 67, 211–225.

Liu, J., Liu, Y., Jin, Y., Li, F., 2022a. A decision variable assortment-based evolutionary algorithm for dominance robust multiobjective optimization. IEEE Transactions on Systems, Man, and Cybernetics: Systems 52, 3360–3375.

Liu, Y., Liu, J., Tan, S., Yang, Y., Li, F., 2022b. A bagging-based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization. Neural Computing and Applications 34, 12097–12118.

Ma, X., Yin, J., Zhu, A., Li, X., Yu, Y., Wang, L., Qi, Y., Zhu, Z., 2022. Enhanced multifactorial evolutionary algorithm with meme helper-tasks. IEEE Transactions on Cybernetics 52, 7837–7851.

Martinez, A.D., Del Ser, J., Osaba, E., Herrera, F., 2022. Adaptive multifactorial evolutionary optimization for multitask reinforcement learning. IEEE Transactions on Evolutionary Computation 26, 233–247.

Min, A.T.W., Ong, Y., Gupta, A., Goh, C.K., 2017. Multiproblem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems. IEEE Transactions on Evolutionary Computation 23, 15–28.

Rice, J., Cloninger, C.R., Reich, T., 1978. Multifactorial inheritance with cultural transmission and assortative mating. i. description and basic properties of the unitary models. American journal of human genetics 30, 618–643.

Rosenblatt, J.D., Benjamini, Y., 2018. On mixture alternatives and wilcoxons signed-rank test. The American Statistician 72, 344–347.

Shang, K., Ishibuchi, H., He, L., Pang, L.M., 2021. A survey on the hypervolume indicator in evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation 25, 1–20.

Tian, Y., 2011. Anode-cathode-distance model for aluminium reduction cell and its energy consumption. Light Metals 2, 31–33.

Wang, B.C., Li, H.X., Fellow, IEEE, Zhang, Q., 2021a. Decomposition-based multiobjective optimization for constrained evolutionary optimization. IEEE Transactions on Systems, Man, and Cybernetics: Systems 51, 574–587.

Wang, C., Liu, J., Wu, K., Wu, Z., 2022. Solving multitask optimization problems with adaptive knowledge transfer via anomaly detection. IEEE Transactions on Evolutionary Computation 26, 304–318.

Wang, H., Feng, L., Jin, Y., Doherty, J., 2021b. Surrogate-assisted evolutionary multitasking for expensive minimax optimization in multiple scenarios. IEEE Computational Intelligence Magazine 16, 34–48.

Yao, L., Ding, W., He, T., Liu, S., Nie, L., 2022. A multiobjective prediction model with incremental learning ability by developing a multi-source filter neural network for the electrolytic aluminium process. Applied Intelligence 52, 17387–17409.

Yao, L., Li, T., Li, Y., Long, W., Yi, J., 2019. An improved feed-forward neural network based on ukf and strong tracking filtering to establish energy consumption model for aluminum electrolysis process. Neural Computing and Applications 31, 4271–4285.

Yi, J., Bai, J., He, H., Peng, J., Tang, D., 2018. ar-moea: A novel preference-based dominance relation for evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation 23, 788–802.

Yi, J., Bai, J., He, H., Zhou, W., Yao, L., 2020. A multifactorial evolutionary algorithm for multitasking under interval uncertainties. IEEE Transactions on Evolutionary Computation 24, 908–922.

Yi, J., Bai, J., Zhou, W., He, H., Yao, L., 2017. Operating parameters optimization for the aluminum electrolysis process using an improved quantum-behaved particle swarm algorithm. IEEE Transactions on Industrial Informatics 14, 3405–3415.

Yi, J., Zhang, W., Bai, J., Zhou, W., Yao, L., 2022. Multifactorial evolutionary algorithm based on improved dynamical decomposition for many-objective optimization problems. IEEE Transactions on Evolutionary Computation 26, 334–348.

Zhang, Q., Hui, L., 2007. Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11, 712–731.

Zitzler, E., Künzli, S., 2004. Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer. pp. 832–842.

Zitzler, E., Laumanns, M., Thiele, L., 2001. Spea2: Improving the strength pareto evolutionary algorithm. TIK-report 103, 1–21.